

# £ironic\$

Bonfatti Andrea Cocchi Giovanni

- A) Descrizione della realtà da modellare
- B) Glossario
- C) Schema scheletro
- D) Progetto concettuale e gestione dei vincoli
- E) Diagramma E-R
- F) Progetto logico
- G) Creazione tabelle
- H) Vincoli
- I) Inserimento dei dati
- J) Operazioni in SQL
- K) Dato derivato
- L) Progetto fisico
- M) Java JDBC

## A) descrizione della realtà da modellare

Si vogliono gestire i dati relativi ad una catena di negozi di elettronica con sedi sul territorio italiano. I punti vendita permettono oltre al classico acquisto di prodotti in presenza, anche l'acquisto online e la consegna a domicilio.

Un **punto vendita** ha un codice, una città, un indirizzo, una partita IVA, un magazzino, un numero di telefono ed una mail.

Non possono esistere due punti vendita nella stessa città per evitare concorrenza, il marchio £ironics viene concesso ad un solo negozio per città.

I punti vendita possono organizzare delle **promozioni settimanali** per i clienti tesserati, in una settimana, un punto vendita può offrire in promozione un solo articolo, ed in una certa settimana, quel dato articolo può essere in promozione in un solo punto vendita. Inoltre un articolo può essere offerto in promozione una sola volta dallo stesso punto vendita.

All'interno del punto vendita lavorano i **dipendenti**.

Per ogni punto vendita, è necessario che un dipendente sia il **responsabile**, e per un responsabile è necessario indicare almeno un sostituto.

Dei dipendenti si memorizza la matricola (univoca all'interno del punto vendita), il nome, il cognome, l'indirizzo email, il numero di cellulare, la data di assunzione e la qualifica.

Un dipendente può lavorare in uno e un solo punto vendita, un responsabile è un responsabile di uno ed un solo punto vendita.

I **clienti** si dividono in clienti fisici (di cui non è dato conoscere nulla in quanto potenzialmente entrano nel punto vendita, acquistano ed escono) e clienti online. Di questi ultimi è necessario memorizzare indirizzo email, password, nome, cognome, numero di cellulare, codice tessera (se presente), e residenza (città, via, civico).

Dei clienti si distinguono anche i clienti **tesserati** o **soci**, clienti con anche un codice di tessera che possono usufruire delle offerte dei punti vendita.

I clienti ordinano **prodotti**. I prodotti sono descritti da un codice, una categoria, marca, prezzo, nome, quantità residua, numero acquisti e fornitore. I clienti possono inoltre lasciare recensioni ai vari prodotti, per consigliare o sconsigliare l'acquisto di un prodotto.

Lo stesso prodotto può essere presente in più punti vendita. I prodotti sono forniti dai **fornitori**, per i quali è necessario memorizzare partita IVA, nome, indirizzo email e recapito telefonico.

I clienti effettuano **ordini** di prodotti. Un ordine è descritto dal **dettaglio** (quantità e prezzo ), dalla mail del cliente che l'ha effettuato e dal punto vendita dal quale parte, e dalla destinazione da raggiungere.

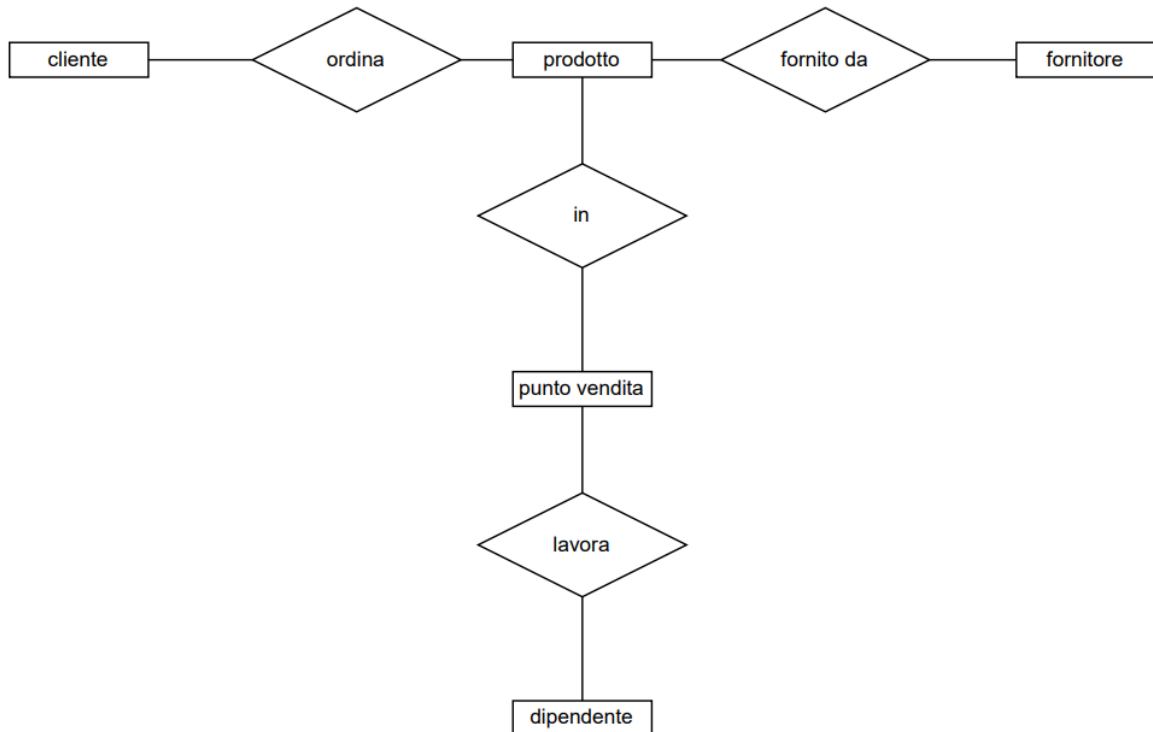
Risulta importante il fatto che un cliente possa ordinare lo stesso prodotto anche più volte nella stessa giornata, se ad esempio si è dimenticato di specificare la quantità nel dettaglio.

## B) Glossario

Termine	Descrizione	Sinonimi	Legami
Punto Vendita	codice_PV indirizzo partita IVA indirizzo email telefono	Negoziò	Dipendente Responsabile Prodotto Presente
Dipendente	Codice Fiscale Matricola nome cognome indirizzo email telefono qualifica Matricola responsabile Codice_PV	lavoratore magazziniere commesso	Punto vendita Persona
Responsabile	Codice Fiscale Matricola Nome Cognome Indirizzo email Telefono Sostituto	capo direttore	Punto vendita Persona
Cliente	Codice Fiscale Email Password Nome Cognome Telefono Città Via Civico	acquirente	Ordine Recensione Socio
Socio	Codice tessera Username	Tesserato	Promozione Cliente
Fornitore	Codice Fiscale Nome Cognome Email Telefono Partita IVA	Venditore	Fornitura Persona

Prodotto	Codice Categoria Marca Prezzo Nome Quantità residua	Articolo Oggetto	Promozione Punto vendita Ordine Fornitura
Ordine	data ora destinazione	Acquisto	Dettaglio Cliente
Persona	Codice Fiscale Nome Cognome Email Telefono		Fornitore Cliente Dipendente Responsabile
Recensione	Codice_Cliente Stelle Titolo Testo Codice_Prodotto		Prodotto Cliente
Promozione	Settimana Codice_Prodotto Codice_PV Prezzo scontato	Sconto Promozione	Socio Prodotto Punto Vendita
Presente	Nome Codice_PV		Citta Punto Vendita
Citta	Nome	Location	Presente
Fornitura	Partita IVA Data		Fornitore Prodotto

### C) Schema scheletro



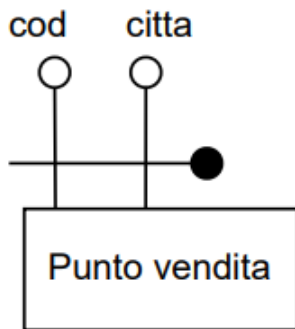
#### Strategia progettuale

Si opta per una strategia **modulare**, iniziando da un primo schema scheletro che descrive solo in modo concettuale la realtà da analizzare, per poi descrivere in modo dettagliato ogni singola parte separatamente, compresi i vincoli, e fondendo il tutto in un unico schema E-R completo.

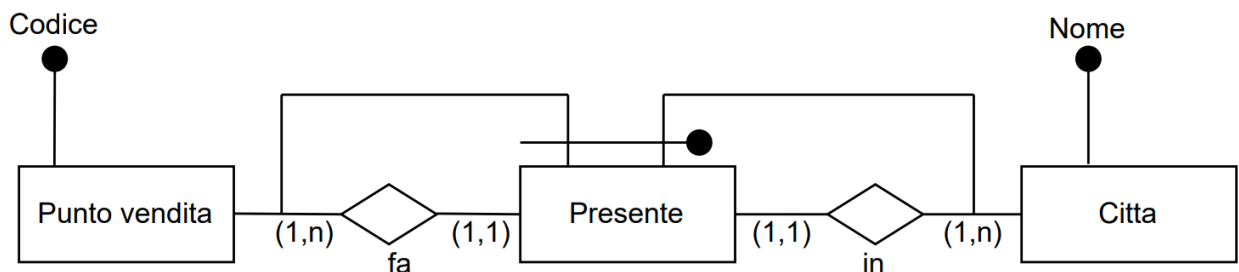
## D) Progetto concettuale e gestione dei vincoli

### D.1) Punto vendita

Il punto vendita deve essere unico all'interno di una città, di conseguenza non ci potranno essere due negozi £ironic\$ all'interno della stessa città, per modellare questo, la chiave composta con codice punto vendita e città è errato in quanto consente di inserire due punti vendita in una stessa città.



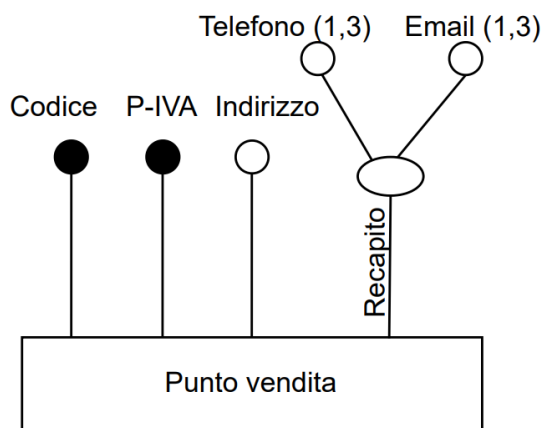
La città va quindi reificata ed associata al punto vendita con una relazione, e non è ancora sufficiente, bisogna introdurre l'entità "presente" che mette in relazione un punto vendita ad una città



Qui un esempio di inserimenti corretti ed inserimenti che violano la chiave dei presente

Punto vendita	Città	Presente
1	Modena	1 Modena OK
2	Bologna	2 Modena OK
2	Milano	2 Milano NO
3	Modena	3 Modena NO
3	Torino	3 Torino OK

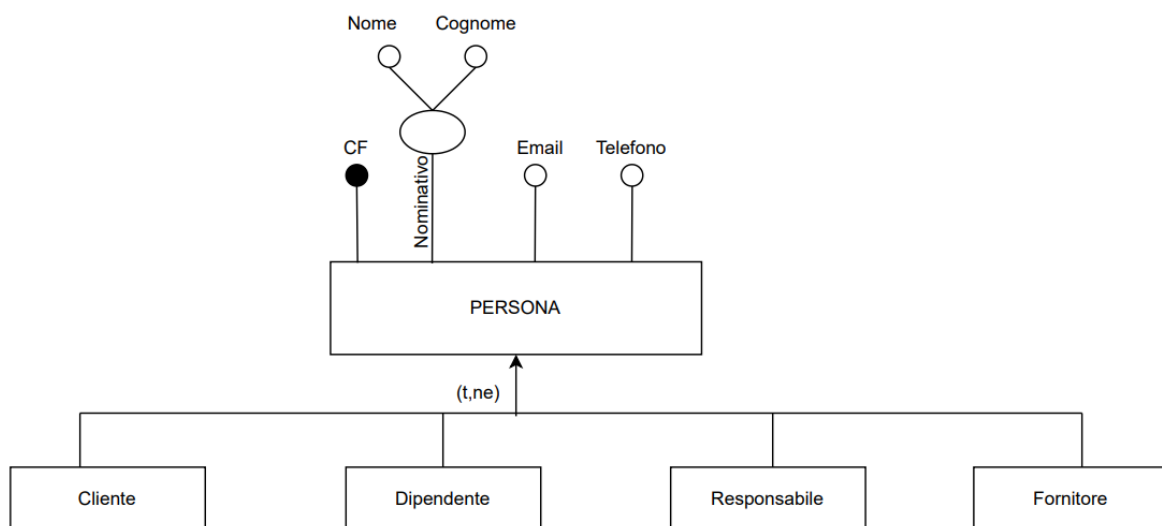
La chiave composta di “Presente” impedisce di registrare più punti vendita nella stessa città, anche se con codici punto vendita differenti



La chiave di punto vendita è il codice punto vendita, e come chiave alternativa la partita iva del punto vendita. Altri attributi sono l'indirizzo ed il recapito.

## D.2) Persona

All'interno del database vengono memorizzate informazioni su varie persone, come i clienti, i fornitori, i dipendenti ed i responsabili. Molte delle informazioni memorizzate per le persone sono le stesse, da qui l'idea di aggiungere la gerarchia persona che raccoglie gli attributi comuni a tutte le persone



I dati comuni alle persone sono codice fiscale, nominativo, indirizzo email e telefono.

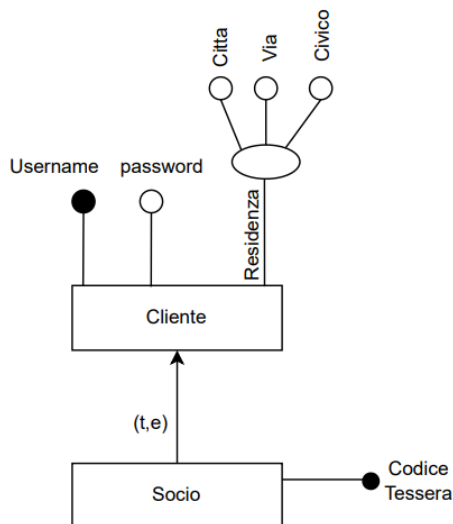
La gerarchia è totale, in quanto ogni elemento di “persona” è in relazione con almeno un elemento delle classi generalizzate, non esclusiva in quanto ogni istanza di “persona” può far parte di più entità generalizzate,



un dipendente ad esempio può anche essere cliente, tanto quanto un fornitore può essere cliente.

### D.2.1) Cliente

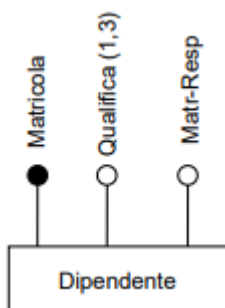
All'interno di cliente ritroviamo tutti i dati di persona, ed introduciamo una nuova gerarchia, in quanto i clienti si dividono in clienti e clienti tesserati o soci.



La gerarchia è di tipo totale ed esclusivo, in quanto i clienti o sono tesserati e quindi soci di £ironic\$, o non lo sono. Una chiave alternativa al codice fiscale per cliente può essere il suo username, anch'esso deve essere univoco, e per i soci può essere una chiave alternativa il codice tessera, anch'esso univoco.

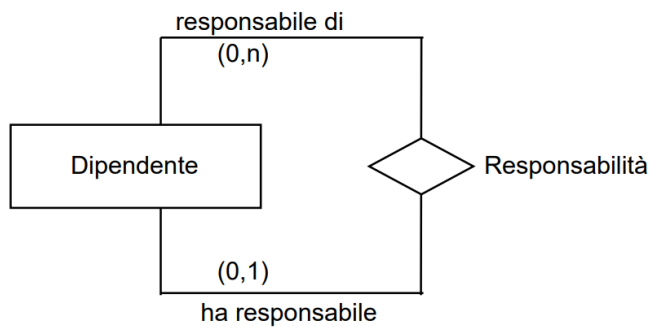
### D.2.2) Dipendente

Anche all'interno di dipendente ritroviamo tutti i dati di persona, in aggiunta abbiamo la matricola che può essere chiave alternativa in quanto anche la matricola deve essere univoca, e la qualifica, supponendo che un dipendente debba avere almeno una qualifica e ne possa avere al massimo 3. In dipendente abbiamo anche il numero di matricola del responsabile.



### D.2.3) Responsabile

Un esempio di autoassociazione all'interno del database poteva essere il responsabile dei dipendenti, fatta nel seguente modo



Ma questo non modella alla perfezione una situazione realistica, in cui il responsabile può essere uno ed un solo, ma ci deve essere sempre una persona incaricata di sostituirlo nel caso di assenza.

Responsabile diventa quindi un'entità anzi che una relazione.



Essendo una specializzazione di "persona", acquisisce gli attributi di persona e si specializza con l'attributo matricola e l'attributo "sostituto", il numero di matricola di chi dovrà occuparsi della sostituzione in caso di assenza

### D.2.4) Fornitore

L'ultima sottoclasse di persona è fornitore, chi fornisce i prodotti ai vari punti vendita



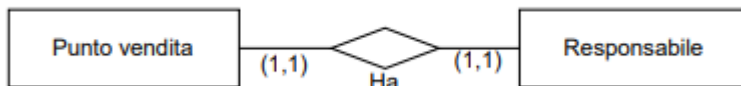
Anche "fornitore" ha tutti gli attributi di persona, e si specializza con la partita iva.

### D.3) Punto Vendita - Persona

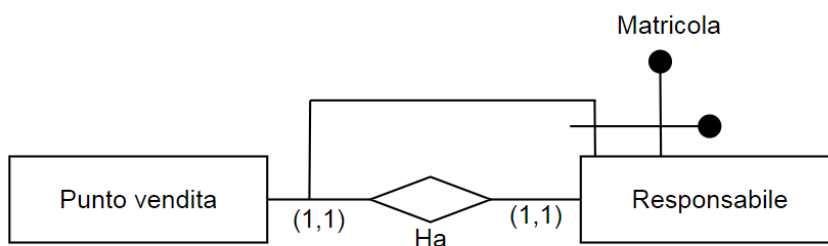
Le entità di persona sono praticamente tutte in relazione con un punto vendita, in particolare

#### D.3.1 ) Punto vendita - Responsabile

In un punto vendita può esserci uno ed un solo responsabile, ed un responsabile è responsabile di uno ed un solo punto vendita



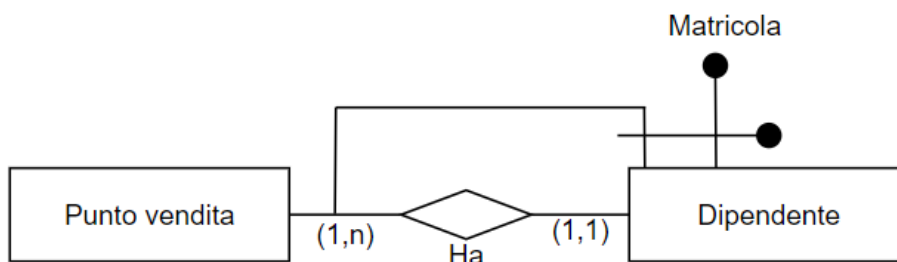
Questo è corretto, ma il responsabile è un'entità che deve essere debole rispetto al punto vendita, in caso di chiusura ed eliminazione del punto vendita, dovrebbe essere eliminato anche il responsabile del punto vendita



La matricola è già chiave di responsabile, la coppia CodicePuntoVendita - Matricola diventa chiave alternativa di responsabile.

#### D.3.2 ) Punto vendita - Dipendente

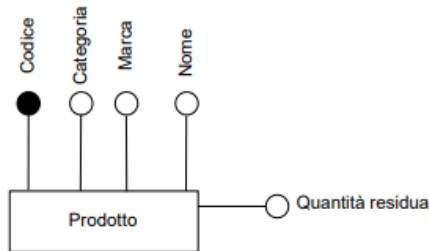
Il medesimo ragionamento viene fatto sui dipendenti, un punto vendita ha più dipendenti, mentre un dipendente lavora in uno ed un solo punto vendita, in caso di chiusura ed eliminazione del punto vendita, dovrebbero essere eliminati anche tutti i dipendenti del punto vendita



Anche in questo caso si crea la chiave alternativa per dipendente formata da CodicePuntoVendita - Matricola.

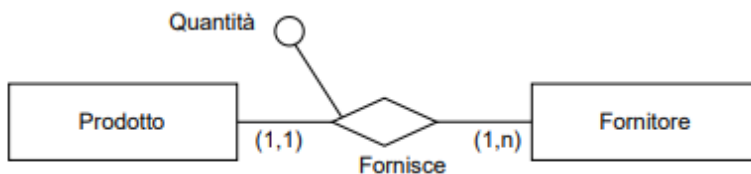
## D.4) Prodotto

I prodotti sono presenti in vari punti vendita, hanno un codice, appartengono ad una categoria, hanno una marca, un nome, un prezzo, una quantità residua e opionalmente una descrizione.

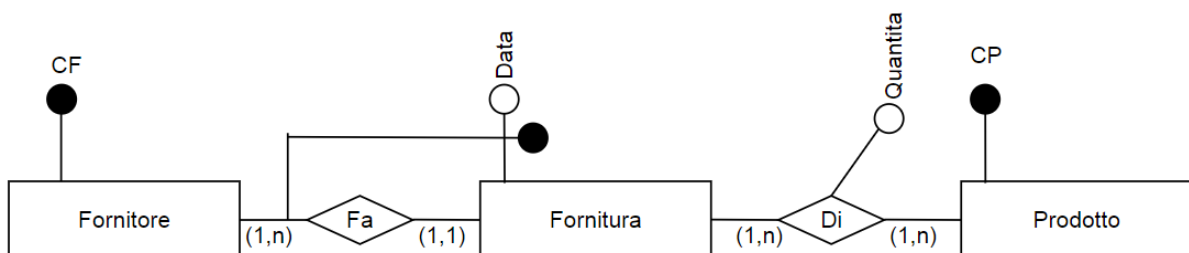


Tutti gli articoli sono disponibili in qualsiasi punto vendita, in quanto vengono selezionati e forniti da £ironic\$, ma in ogni punto vendita possono avere un prezzo diverso.

I fornitori forniscono i prodotti che poi vengono ordinati dai clienti.



Questo primo modello però non è errato, in quanto un fornitore potrebbe fornire un prodotto una sola volta, occorre reificare "fornisce" e farla diventare un'entità



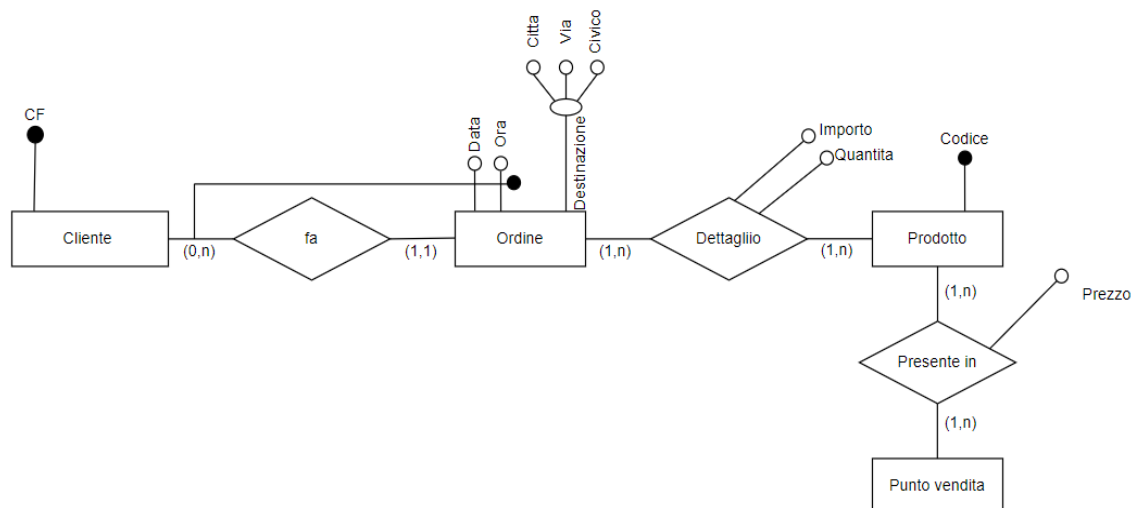
In questo modo un fornitore può fare da 1 a n forniture, una fornitura è fatta da uno ed un solo fornitore, in una certa data, un fornitore fa una fornitura di un certo prodotto, se la data cambia lo stesso fornitore può fare nuovamente la stessa fornitura dello stesso prodotto. forniture.

### D.5) Ordine e dettaglio

Gli acquisti possono avvenire sia “in presenza” che attraverso ordini. Per quello che riguarda un acquisto fatto di persona non è dato sapere nulla del cliente, l'unica cosa che è importante fare a livello di DB è aggiornare la quantità residua dei prodotti acquistati. Per quello che riguarda gli acquisti on-line invece ci sono molte più informazioni che si devono raccogliere.

Un ordine è sicuramente effettuato da un solo cliente, la chiave composta Data-Ora-CF garantisce che ogni ordine di ogni cliente sia unico, l'ordine debole rispetto a cliente verrà poi gestito in SQL per la possibilità di eliminare Cliente.

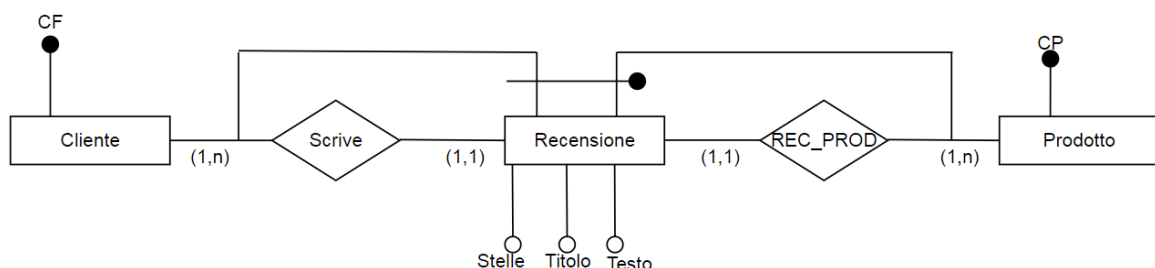
Un ordine ha associato il proprio dettaglio, dove viene specificata la quantità di ogni prodotto e l'importo. Per il prezzo di un prodotto è necessario conoscere anche il punto vendita dal quale l'ordine viene preso in carico ed effettuato.



### D.6) Recensione

I clienti possono scrivere recensioni sugli articoli, così da consigliare o sconsigliare il prodotto ad altri clienti.

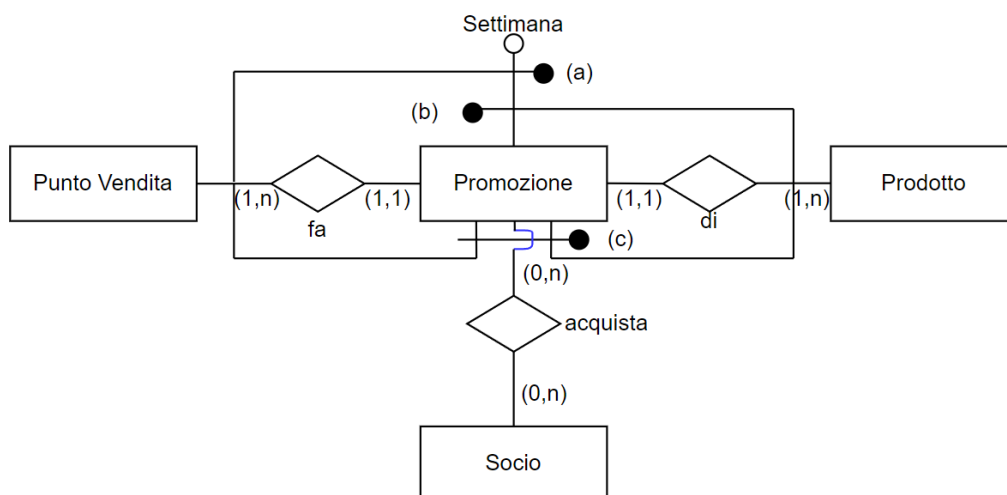
Una recensione può essere scritta da un cliente una sola volta per prodotto, è caratterizzata dal numero di stelle (da 1 a 5), dal titolo e dal testo.



## D.7) Promozione

Le promozioni non sono acquisti normali, ma acquisti riservati ai soli clienti tesserati, e quindi vengono trattate in maniera diversa dagli ordini.

I vincoli da modellare in questo caso sono tre.



**a)** In una settimana, un punto vendita può offrire in promozione un solo prodotto, la chiave di promozione composta da “settimana” e dalla chiave di punto vendita, che si associa (1,1) con prodotto modella questa specifica.

**b)** Nella stessa settimana, lo stesso articolo può essere in promozione in un solo punto vendita, la chiave di promozione formata da “settimana” e dal codice prodotto, che si associa (1,1) con punto vendita modella questa specifica.

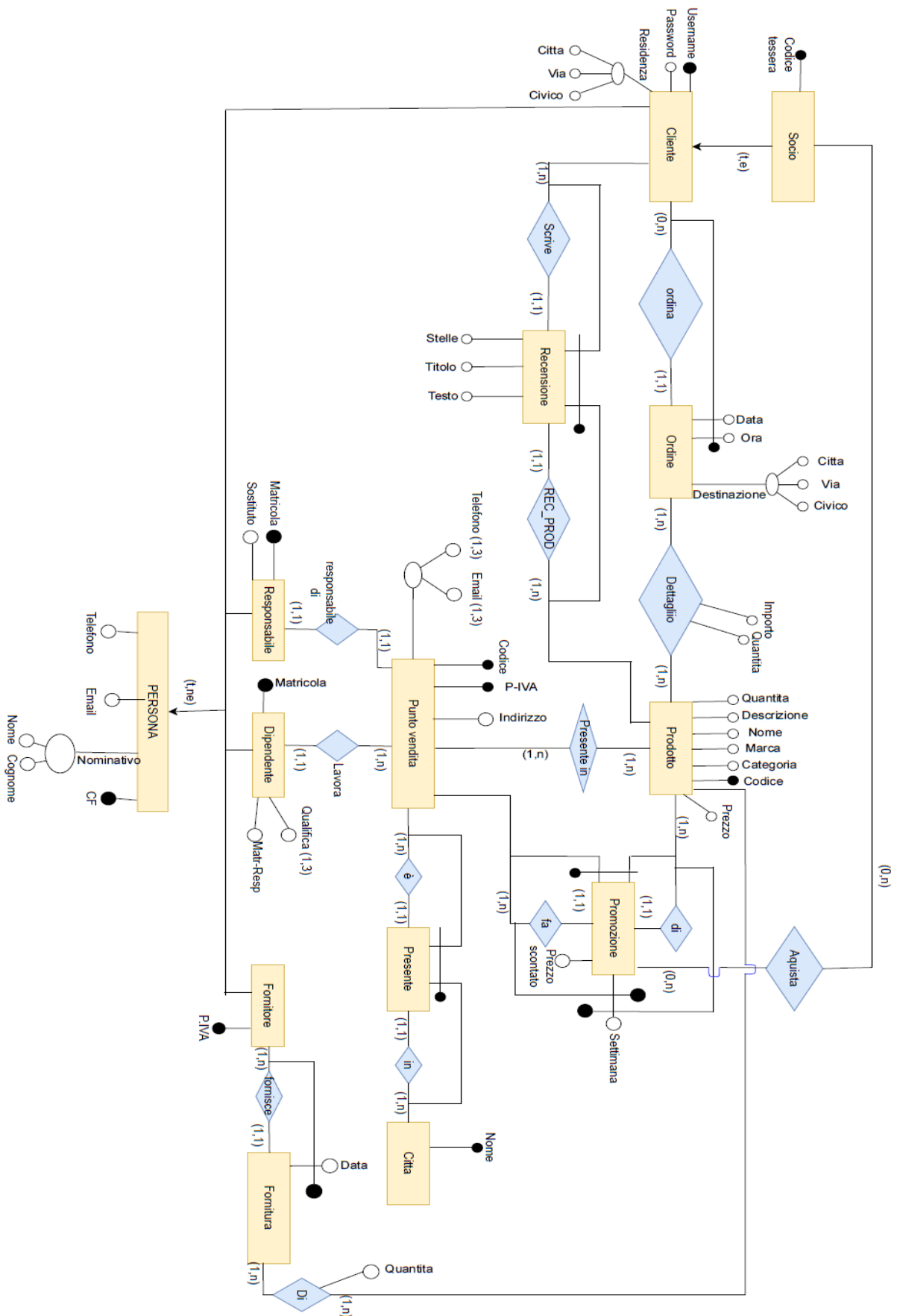
**c)** Un prodotto può essere offerto in promozione una sola volta dallo stesso punto vendita, la chiave di promozione formata dalla chiave di punto vendita e di prodotto rende possibile l'inserimento di un prodotto in offerta in un punto vendita una sola volta.

Di seguito riportati alcuni esempi di “inserimenti” consentiti e altri che violano le chiavi.

C.Punto vendita	C.Prodotto	Settimana	Promozione
A	AX10	1	1 AX10 1 OK
A	MN20	1	A MN20 1 NO
A	MN20	2	A MN20 2 OK

B	AX10	1	B AX10 1 NO
B	AX10	2	B AX10 2 OK
A	AX10	2	A 2 AX10 NO

## E) SCHEMA E-R





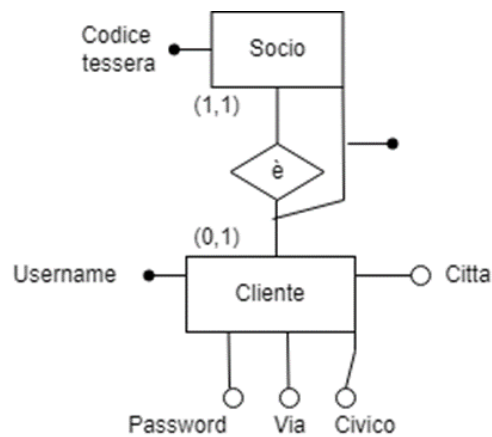
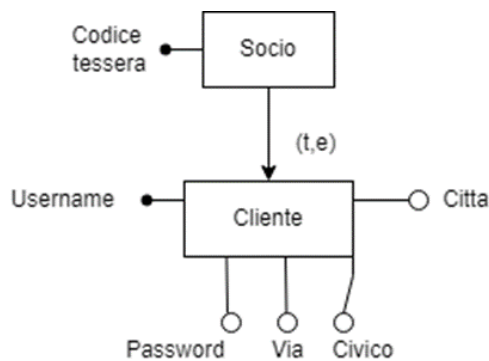
## F) PROGETTO LOGICO

### F.1) Eliminazione delle gerarchie

Per realizzare il progetto logico si è proceduto con l'eliminazione delle gerarchie.

#### F.1.1) Socio

Per la gerarchia Socio => Cliente si sono mantenute distinte le due entità inserendo la chiave alternativa "Username" all'interno di Socio, e si scompone l'attributo composto in 3 attributi semplici



SOCIO (COD\_TESSERA, USERNAME)

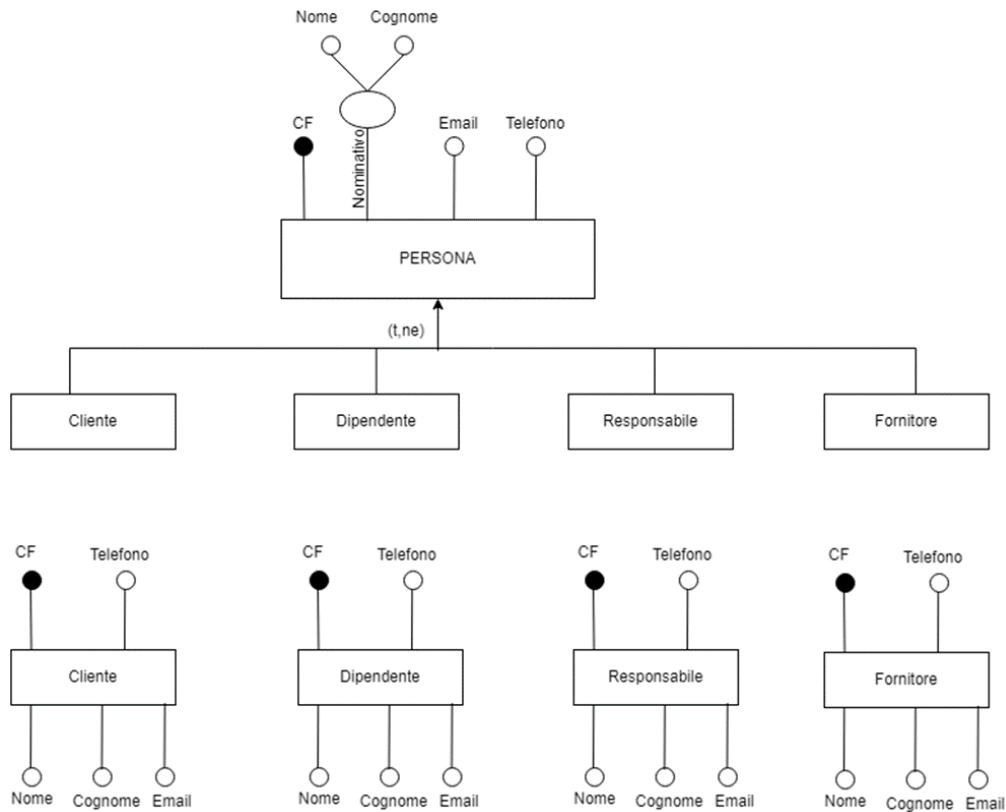
AK: USERNAME

CLIENTE(USERNAME, PASSWORD, CITTA, VIA, CIVICO,  
TELEFONO, CF, EMAIL, NOME, COGNOME)

AK: CF

### F.1.2) Persona

Per la gerarchia di Persona, si è optato per un collasso verso il basso, andando così ad eliminare l'entità persona, distribuendone gli attributi per ogni singola entità figlia



CLIENTE (CF, NOME, COGNOME, EMAIL, TELEFONO)

DIPENDENTE (CF, NOME, COGNOME, EMAIL, TELEFONO)

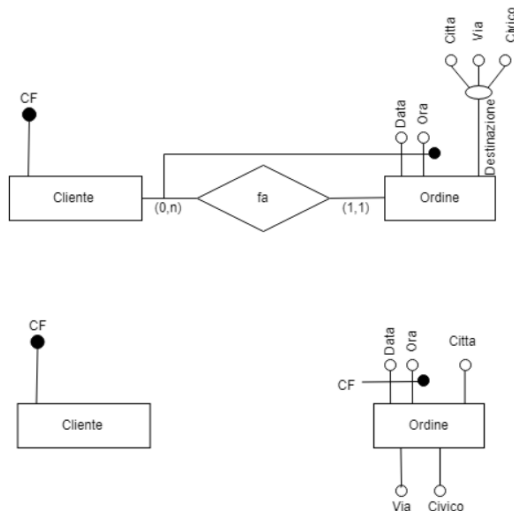
RESPONSABILE (CF, NOME, COGNOME, EMAIL, TELEFONO)

FORNITORE(CF, NOME, COGNOME, EMAIL, TELEFONO)

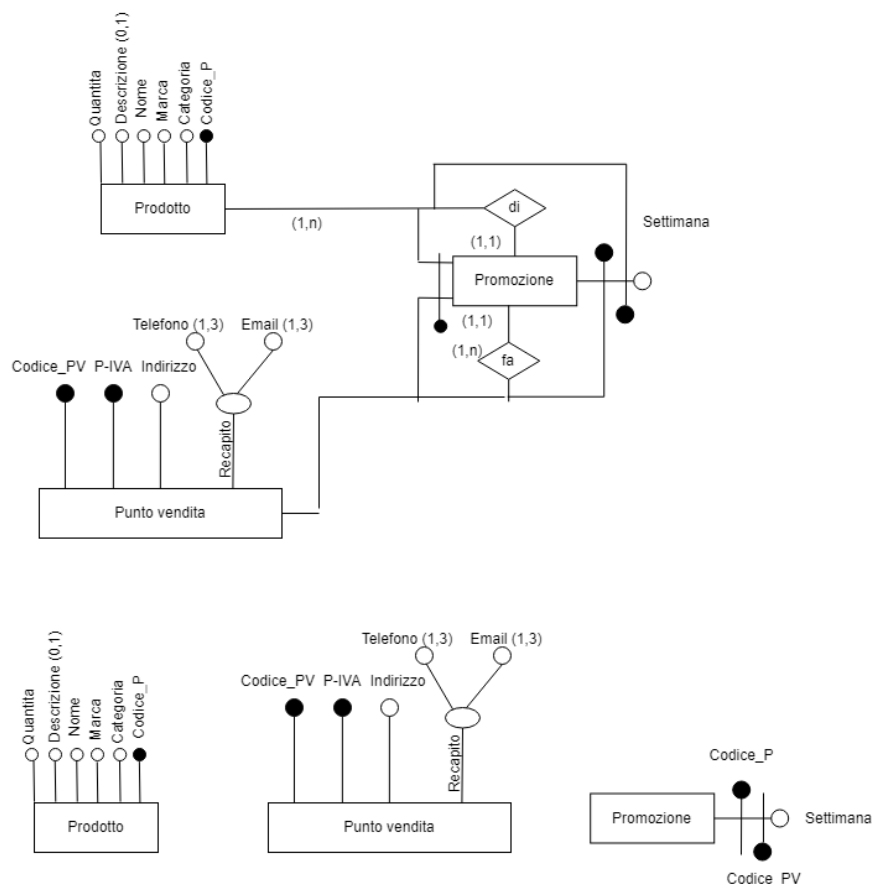
## F.2 ) Eliminazione degli identificatori esterni

Si vanno ora ad eliminare gli identificatori esterni, portando la chiave nell'entità debole

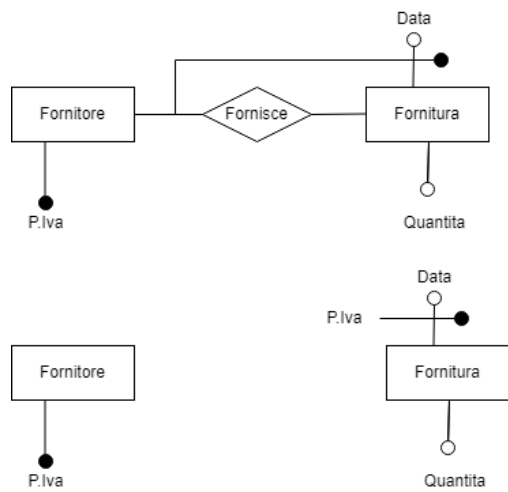
### F.2.1) Cliente -> Ordine



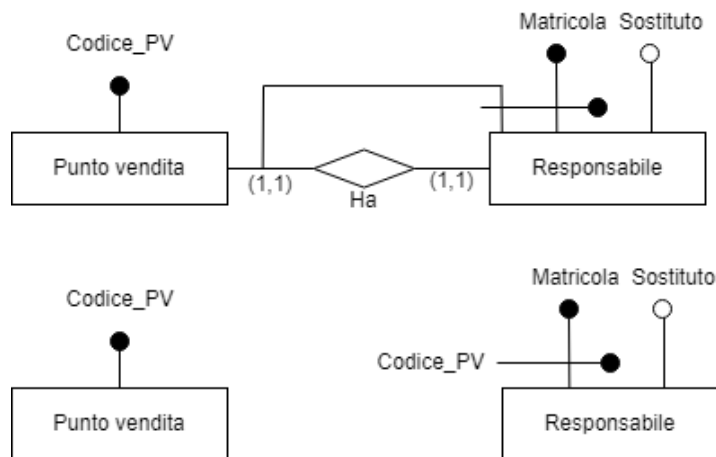
### F.2.2) Prodotto, Punto vendita $\Rightarrow$ promozione



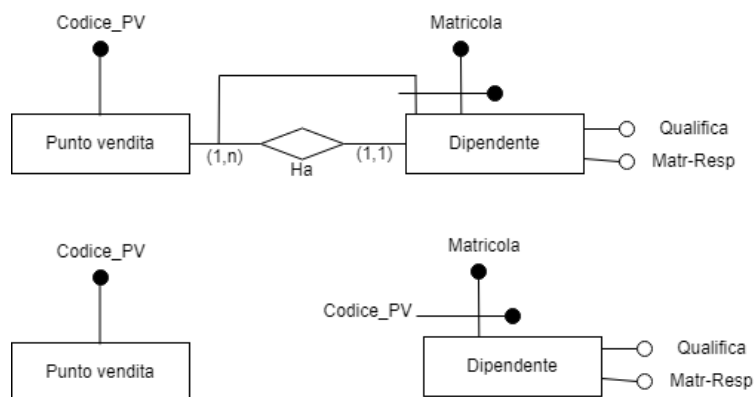
### F.2.3) Fornitore $\Rightarrow$ fornitura



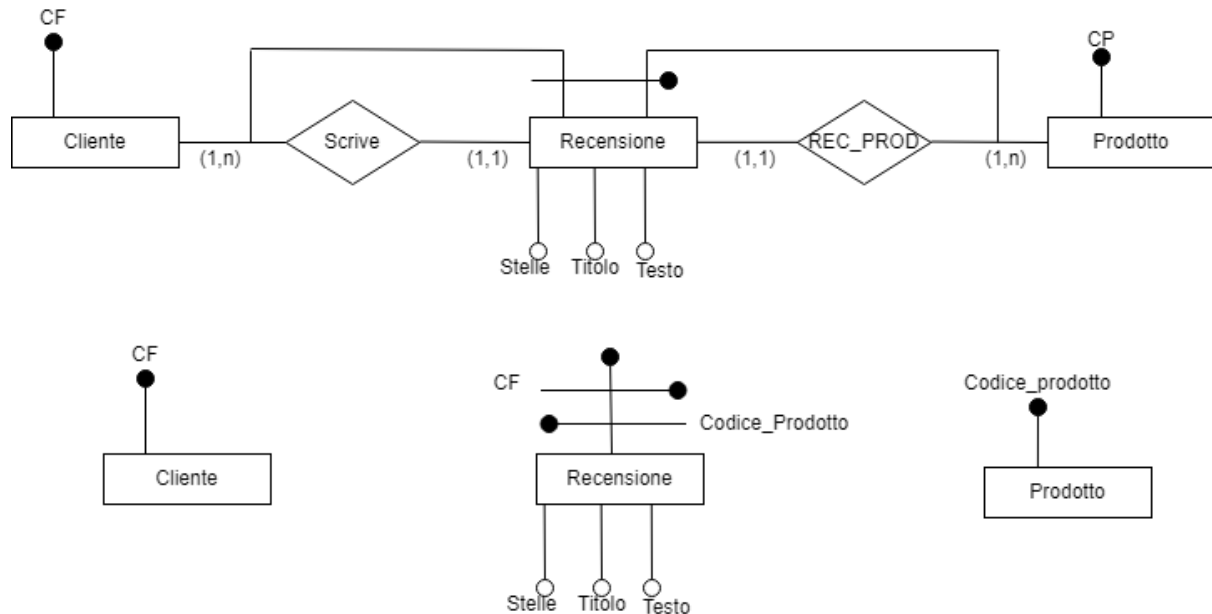
### F.2.4) Punto vendita $\Rightarrow$ responsabile



### F.2.5) Punto vendita $\Rightarrow$ Dipendente



## F.2.6) Prodotto, Cliente ⇒ Recensione



## F.3) Progetto logico completo

**SOCIO**(COD\_TESSERA, USERNAME)

AK: USERNAME

**CLIENTE** (USERNAME, PASSWORD, CITTA, VIA, CIVICO, TELEFONO, CF, EMAIL, NOME, COGNOME)

AK: CF

**ORDINE** (USERNAME, DATA, ORA, CITTA\_DEST, VIA, CIVICO)

FK: USERNAME REFERENCES CLIENTE

**PRODOTTO**(CODICE\_PRODOTTO, QUANTITA\_RESIDUA, DESCRIZIONE, NOME, MARCA, CATEGORIA, PREZZO)

**DETTAGLIO**(USERNAME, DATA, ORA, CODICE\_PRODOTTO, IMPORTO, QUANTITA)

FK: USERNAME, DATA, ORA REFERENCES ORDINE

FK: CODICE\_PRODOTTO REFERENCES PRODOTTO

**RECENSIONE**(USERNAME, CODICE\_PRODOTTO, STELLE, TITOLO, TESTO)

FK: USERNAME REFERENCES CLIENTE

FK: CODICE\_PRODOTTO REFERENCES PRODOTTO

**PUNTO\_VENDITA**( CODICE\_PV, P.IVA, INDIRIZZO, TELEFONO, EMAIL)

AK: PIVA

**PROMOZIONE**(CODICE\_PRODOTTO, CODICE\_PV, SETTIMANA, SCONTO)

AK: CODICE\_PRODOTTO, SETTIMANA

AK: CODICE\_PV, SETTIMANA

FK: CODICE\_PRODOTTO REFERENCES PRODOTTO

FK: CODICE\_PV REFERENCES PUNTO\_VENDITA

**FORNITORE**(P.IVAF, CF, TELEFONO, EMAIL, NOME, COGNOME)

AK: CF

**FORNITURA**(PIVAF, DATA)

FK: PIVAF REFERENCES FORNITORE

**DI** (PIVAF, DATA, CODICE\_PRODOTTO, QUANTITA)

FK: PIVAF, DATA REFERENCES FORNITURA

FK: CODICE\_PRODOTTO REFERENCES PRODOTTO

**PRESENTE\_IN**(CODICE\_PV, CODICE\_PRODOTTO)

FK CODICE\_PV REFERENCES PUNTO VENDITA

FK CODICE\_PRODOTTO REFERENCES PRODOTTO

**ACQUISTA**(COD\_TESSERA, CODICE\_PV, CODICE\_PRODOTTO, PREZZO\_SCONTATO)

FK: COD\_TESSERA REFERENCES SOCIO

FK: CODICE\_PV, CODICE\_PRODOTTO REFERENCES

PROMOZIONE

**CITTA**(NOME)

**PRESENTE**(NOME, CODICE PV)

FK: NOME REFERENCES CITTA

FK: CODICE\_PV REFERENCES PUNTO\_VENDITA

**RESPONSABILE**(MATRICOLA\_RESPONSABILE, CF, SOSTITUTO, TELEFONO, EMAIL, NOME, COGNOME)

FK: CODICE\_PV REFERENCES PUNTO\_VENDITA

AK: CF

**RESPONSABILE\_DI**(MATRICOLA\_RESPONSABILE, CODICE PV)

FK: MATRICOLA\_RESPONSABILE REFERENCES  
RESPONSABILE

FK: CODICE\_PV REFERENCES PUNTO\_VENDITA

**DIPENDENTE**(MATRICOLA\_DIPENDENTE, QUALIFICA, CF, TELEFONO, EMAIL, NOME, COGNOME)

AK: CF

FK: CODICE\_PV REFERENCES PUNTO\_VENDITA

LAVORA(MATRICOLA\_DIPENDENTE, CODICE\_PV)

FK: MATRICOLA\_DIPENDENTE REFERENCES DIPENDENTE

FK: CODICE\_PV REFERENCES PUNTO\_VENDITA

## **G) Creazione tabelle**

```
CREATE TABLE SOCIO (  
  COD_TESSERA CHAR (10) PRIMARY KEY,  
  USERNAME VARCHAR (100) NOT NULL UNIQUE  
);
```

```
CREATE TABLE CLIENTE (  
  USERNAME VARCHAR(100) PRIMARY KEY,  
  PASSWORD VARCHAR(25) NOT NULL,  
  CITTA VARCHAR(50),  
  VIA VARCHAR(100),  
  CIVICO SMALLINT,  
  CF CHAR(16) NOT NULL UNIQUE,  
  EMAIL VARCHAR(40) UNIQUE,  
  NOME VARCHAR(100),  
  COGNOME VARCHAR(100)  
);
```

```
CREATE TABLE ORDINE (  
  USERNAME VARCHAR(100),  
  DATA DATE,  
  ORA TIME(0),  
  CITTA_DEST VARCHAR(100),  
  VIA VARCHAR(100) NOT NULL,  
  CIVICO SMALLINT NOT NULL,  
  PRIMARY KEY(USERNAME, DATA, ORA),  
  FOREIGN KEY(USERNAME) REFERENCES CLIENTE  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE PRODOTTO (  
  CODICE_PRODOTTO CHAR(10) PRIMARY KEY,  
  QUANTITA_RESIDUA INTEGER NOT NULL,  
  DESCRIZIONE VARCHAR(200) NOT NULL,  
  NOME VARCHAR(100) NOT NULL,  
  MARCA VARCHAR(100),
```



```
CATEGORIA VARCHAR(50),  
PREZZO REAL  
        CHECK (PREZZO>0)  
);
```

```
CREATE TABLE DETTAGLIO (  
    USERNAME VARCHAR(100),  
    DATA DATE,  
    ORA TIME(0),  
    CODICE_PRODOTTO CHAR(10),  
    IMPORTO REAL NOT NULL  
        CHECK (IMPORTO >0),  
    QUANTITA INT NOT NULL  
    CHECK (QUANTITA>0),  
    PRIMARY KEY(USERNAME, DATA, ORA, CODICE_PRODOTTO),  
    FOREIGN KEY(USERNAME, DATA, ORA) REFERENCES ORDINE  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(CODICE_PRODOTTO) REFERENCES PRODOTTO  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE RECENSIONE (  
    USERNAME VARCHAR(100),  
    CODICE_PRODOTTO CHAR(10),  
    STELLE SMALLINT NOT NULL  
    CHECK (STELLE BETWEEN 1 AND 5),  
    TITOLO VARCHAR(20),  
    TESTO VARCHAR(500),  
    PRIMARY KEY (USERNAME,CODICE_PRODOTTO),  
    FOREIGN KEY (CODICE_PRODOTTO) REFERENCES PRODOTTO  
    ON DELETE NO ACTION ON UPDATE CASCADE,  
    FOREIGN KEY (USERNAME) REFERENCES CLIENTE  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE PUNTO_VENDITA (  
  CODICE_PV CHAR(5) PRIMARY KEY,  
  PIVA CHAR(11) NOT NULL UNIQUE,  
  INDIRIZZO VARCHAR(150) NOT NULL,  
  TELEFONO VARCHAR(20),  
  EMAIL VARCHAR(40)  
);
```

```
CREATE TABLE PROMOZIONE (  
  CODICE_PRODOTTO CHAR(10) NOT NULL,  
  CODICE_PV CHAR(5) NOT NULL,  
  SETTIMANA INT NOT NULL,  
  PREZZO_SCONTATO REAL,  
  PRIMARY KEY(CODICE_PRODOTTO, CODICE_PV),  
  FOREIGN KEY (CODICE_PRODOTTO) REFERENCES PRODOTTO  
  ON DELETE NO ACTION ON UPDATE CASCADE,  
  FOREIGN KEY(CODICE_PV) REFERENCES PUNTO_VENDITA  
  ON DELETE NO ACTION ON UPDATE CASCADE,  
  UNIQUE(CODICE_PRODOTTO,SETTIMANA),  
  UNIQUE(CODICE_PV,SETTIMANA)  
);
```

```
CREATE TABLE FORNITORE (  
  PIVAF CHAR(11) PRIMARY KEY,  
  CF CHAR(16) NOT NULL UNIQUE,  
  EMAIL VARCHAR(40) UNIQUE,  
  NOME VARCHAR(100),  
  COGNOME VARCHAR(100),  
  TELEFONO VARCHAR(10)  
);
```

```
CREATE TABLE FORNITURA (  
  PIVAF CHAR(11),  
  DATA DATE,  
  PRIMARY KEY( PIVAF, DATA),  
  FOREIGN KEY (PIVAF) REFERENCES FORNITORE  
  ON DELETE CASCADE ON UPDATE CASCADE
```

);

```
CREATE TABLE DI (  
PIVAF CHAR(11),  
DATA DATE,  
CODICE_PRODOTTO CHAR(10),  
QUANTITA INT,  
PRIMARY KEY(PIVAF, DATA, CODICE_PRODOTTO),  
FOREIGN KEY (PIVAF, DATA) REFERENCES FORNITURA  
ON DELETE NO ACTION ON UPDATE CASCADE,  
FOREIGN KEY (CODICE_PRODOTTO) REFERENCES PRODOTTO  
ON DELETE NO ACTION ON UPDATE CASCADE  
);
```

```
CREATE TABLE PRESENTE_IN (  
CODICE_PV CHAR(5),  
CODICE_PRODOTTO CHAR(10),  
PRIMARY KEY (CODICE_PV, CODICE_PRODOTTO),  
FOREIGN KEY (CODICE_PV) REFERENCES PUNTO_VENDITA  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (CODICE_PRODOTTO) REFERENCES PRODOTTO  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE ACQUISTA (  
COD_TESSERA CHAR (10),  
CODICE_PV CHAR(5),  
CODICE_PRODOTTO CHAR(10),  
PRIMARY KEY (COD_TESSERA, CODICE_PV,  
CODICE_PRODOTTO),  
FOREIGN KEY (COD_TESSERA) REFERENCES SOCIO  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (CODICE_PV) REFERENCES PUNTO_VENDITA  
ON DELETE NO ACTION ON UPDATE CASCADE,  
FOREIGN KEY (CODICE_PRODOTTO) REFERENCES PRODOTTO  
ON DELETE NO ACTION ON UPDATE CASCADE  
);
```

```
CREATE TABLE CITTA (  
NOME VARCHAR (100) PRIMARY KEY  
);
```

```
CREATE TABLE PRESENTE (  
NOME VARCHAR(100),  
CODICE_PV CHAR(5),  
PRIMARY KEY (NOME, CODICE_PV),  
FOREIGN KEY (NOME) REFERENCES CITTA  
ON DELETE NO ACTION ON UPDATE CASCADE,  
FOREIGN KEY (CODICE_PV) REFERENCES PUNTO_VENDITA  
ON DELETE NO ACTION ON UPDATE CASCADE  
);
```

```
CREATE TABLE RESPONSABILE (  
MATRICOLA_RESPONSABILE CHAR(6) PRIMARY KEY,  
CF CHAR(16) NOT NULL UNIQUE,  
EMAIL VARCHAR(40) UNIQUE,  
NOME VARCHAR(100),  
COGNOME VARCHAR(100),  
SOSTITUTO CHAR(6) NOT NULL,  
TELEFONO VARCHAR(20)  
);
```

```
CREATE TABLE DIPENDENTE (  
MATRICOLA_DIPENDENTE CHAR(6) PRIMARY KEY,  
QUALIFICA VARCHAR(100) NOT NULL,  
CF CHAR(16) NOT NULL UNIQUE,  
EMAIL VARCHAR(40) UNIQUE,  
NOME VARCHAR(100),  
COGNOME VARCHAR(100),  
TELEFONO VARCHAR(20)  
);
```

```

CREATE TABLE LAVORA (
MATRICOLA_DIPENDENTE CHAR(6),
CODICE_PV CHAR(5),
PRIMARY KEY (MATRICOLA_DIPENDENTE, CODICE_PV),
FOREIGN KEY (MATRICOLA_DIPENDENTE) REFERENCES
DIPENDENTE
    ON DELETE CASCADE ON UPDATE CASCADE ,
FOREIGN KEY (CODICE_PV) REFERENCES PUNTO_VENDITA
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE RESPONSABILE_DI (
    MATRICOLA_RESPONSABILE CHAR(6) PRIMARY KEY,
    CODICE_PV CHAR (5) NOT NULL UNIQUE,
    FOREIGN KEY (MATRICOLA_RESPONSABILE) REFERENCES
RESPONSABILE
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CODICE_PV) REFERENCES PUNTO_VENDITA
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

## H) Vincoli

Per gestire alcuni vincoli non gestibili da E-R si ricorre a Trigger o Procedure che mantengono coerenza nei dati.

### H.1) /\* FUNZIONE PER L'ACQUISTO DA PARTE DI UN CLIENTE FISICO

**VIENE SCALATA LA QUANTITA DI PRODOTTO ACQUISTATA DALLE SCORTE DI LIRONICS \*/**

```

CREATE FUNCTION ACQUISTO_PRESENZA (COD_PROD CHAR,
QUANT INTEGER)
RETURNS INTEGER
AS $$

```

```

DECLARE
    QAVIABLE INTEGER;

```

```

QUPDATE INTEGER;

BEGIN
  SELECT QUANTITA_RESIDUA INTO QAVIABLE
  FROM PRODOTTO
  WHERE CODICE_PRODOTTO = COD_PROD;

  QUPDATE := QAVIABLE - QUANT;

  UPDATE PRODOTTO
  SET QUANTITA_RESIDUA = QUPDATE
  WHERE CODICE_PRODOTTO = COD_PROD;

  RETURN QUPDATE;
END;
$$ LANGUAGE 'plpgsql';

```

**H.2) /\* TRIGGER CHE NON CONSENTE DI AGGIUNGERE UN DETTAGLIO SE UN PRODOTTO NON E' PRESENTE O SE LA QUANTITA' RICHIESTA SUPERA LA QUANTITA' DISPONIBILE \*/**

```

CREATE FUNCTION UPDATE_ORDINE ()
RETURNS trigger
AS $$

  DECLARE QDISP INTEGER; /* quantità disponibile */
  DECLARE QRES INTEGER; /* quantità residua dopo
l'aggiornamento */
BEGIN
  SELECT QUANTITA_RESIDUA INTO QDISP
  FROM PRODOTTO
  WHERE PRODOTTO.CODICE_PRODOTTO =
NEW.CODICE_PRODOTTO;

  IF QDISP = 0

```

```

        THEN RAISE EXCEPTION 'Articolo momentaneamente non
disponibile';
    END IF;

    QRES := QDISP - NEW.QUANTITA;

    IF QRES < 0
        THEN RAISE EXCEPTION 'Quantità disponibile insufficiente per l'
ordine' ;
    ELSE
        UPDATE PRODOTTO
        SET QUANTITA_RESIDUA = QRES
        WHERE PRODOTTO.CODICE_PRODOTTO =
NEW.CODICE_PRODOTTO;
    END IF;
    RETURN NEW;
END
$$ LANGUAGE 'plpgsql';

```

```

CREATE TRIGGER UPDATE_ORDINE BEFORE
INSERT ON DETTAGLIO FOR EACH ROW
EXECUTE PROCEDURE UPDATE_ORDINE();

```

### **H.3) /\* TRIGGER CHE CONTROLLA CHE L'IMPORTO NEL DETTAGLIO SIA UGUALE ALLA QUANTITA PER IL PREZZO \*/**

```

CREATE FUNCTION CONTROLLA_IMPORTO ()
RETURNS TRIGGER
AS $$
    DECLARE PR REAL;
    DECLARE TOT REAL;
BEGIN

    PR := (SELECT PREZZO
    FROM PRODOTTO
    WHERE CODICE_PRODOTTO = NEW.CODICE_PRODOTTO);

```

```

TOT := NEW.QUANTITA * PR;
IF NEW.IMPORTO < TOT
    THEN RAISE EXCEPTION 'IMPORTO INFERIORE AL DOVUTO';
END IF;
IF NEW.IMPORTO > TOT
    THEN RAISE EXCEPTION 'IMPORTO SUPERIORE AL DOVUTO';
END IF;
RETURN NEW;
END
$$ LANGUAGE 'plpgsql';
CREATE TRIGGER CONTROLLA_IMPORTO BEFORE
INSERT ON DETTAGLIO FOR EACH ROW
EXECUTE PROCEDURE CONTROLLA_IMPORTO();

```

**H.4) /\* TRIGGER CHE AGGIORNA LA QUANTITA DISPONIBILE DI UN PRODOTTO QUANDO VIENE FATTA UNA FORNITURA SE E SOLO SE IL PRODOTTO ESISTE NELLA TABELLA DEI PRODOTTI E SE LA QUANTITA FORNITA MAGGIORE DI 0 \*/**

```

CREATE FUNCTION NUOVA_FORNITURA()
RETURNS TRIGGER
AS $$
BEGIN

    IF NEW.QUANTITA < 0
        THEN RAISE EXCEPTION 'Quantità fornita minore di 0';
    END IF;
    IF NEW.QUANTITA = 0
        THEN RAISE EXCEPTION 'Quantità fornita nulla ';
    END IF;

    UPDATE PRODOTTO
    SET QUANTITA_RESIDUA = QUANTITA_RESIDUA +
NEW.QUANTITA
    WHERE CODICE_PRODOTTO = NEW.CODICE_PRODOTTO;

```



```
    RETURN NEW;  
END  
$$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER NUOVA_FORNITURA BEFORE  
INSERT ON DI FOR EACH ROW  
EXECUTE PROCEDURE NUOVA_FORNITURA();
```

**H.5) /\* TRIGGER CHE CONTROLLA CHE IL PREZZO DI UNA  
PROMOZIONE SIA MINORE DEL PREZZO CHE SOLITAMENTE HA  
L'ARTICOLO \*/**

```
CREATE FUNCTION CONTROLLA_OFFERTA()  
RETURNS TRIGGER  
AS $$
```

```
    DECLARE PR REAL;  
BEGIN  
    IF NEW.PREZZO_SCONTATO < 0  
        THEN RAISE EXCEPTION 'il prezzo dell offerta è minore di 0';  
    END IF;  
    IF NEW.PREZZO_SCONTATO = 0  
        THEN RAISE EXCEPTION 'il prezzo dellofferta non può essere 0 ';  
    END IF;
```

```
    PR:=( SELECT PREZZO  
    FROM PRODOTTO  
    WHERE PRODOTTO.CODICE_PRODOTTO =  
NEW.CODICE_PRODOTTO);
```

```
    IF (NEW.PREZZO_SCONTATO > PR)  
        THEN RAISE EXCEPTION 'il prezzo scontato non può essere  
maggiore del prezzo del prodotto';  
    END IF;  
    RETURN NEW;  
END  
$$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER CONTROLLA_OFFERTA BEFORE  
INSERT ON PROMOZIONE FOR EACH ROW  
EXECUTE PROCEDURE CONTROLLA_OFFERTA();
```

**H.6) /\* TRIGGER CHE NON CONSENTE DI INSERIRE UNA  
RECENSIONE DI UN PRODOTTO SE NON E' STATO ORDINATO \*/**

```
CREATE FUNCTION OK_RECENSIONE()  
RETURNS TRIGGER  
AS $$
```

```
DECLARE US VARCHAR(100);  
DECLARE CO CHAR(10);
```

```
BEGIN  
    US := NEW.USERNAME;  
    CO := NEW.CODICE_PRODOTTO;  
  
    IF NOT EXISTS  
        ((SELECT *  
         FROM DETTAGLIO  
         WHERE USERNAME = US AND CODICE_PRODOTTO = CO))  
  
        THEN RAISE EXCEPTION 'IL CLIENTE NON HA ORDINATO  
QUESTO PRODOTTO, IMPOSSIBILE INSERIRE RECENSIONE';  
    END IF;  
  
    RETURN NEW;  
END  
$$ LANGUAGE 'plpgsql';  
CREATE TRIGGER OK_RECENSIONE BEFORE  
INSERT ON RECENSIONE FOR EACH ROW  
EXECUTE PROCEDURE OK_RECENSIONE();
```

## H.7) Inserimenti errati che causano un eccezione

### H.7.1) /\*INSERIMENTO PER TRIGGER RECENSIONE\*/

INSERT INTO RECENSIONE VALUES ('Fillo93','Bigice3000 ', '3 ',  
'Ottimo prodotto ',' Questo tablet va che è una meraviglia');

#### Messages

```
ERROR: ERRORE:  IL CLIENTE NON HA ORDINATO QUESTO PRODOTTO,  
IMPOSSIBILE INSERIRE RECENSIONE  
CONTEXT:  funzione PL/pgSQL ok_recensione() riga 15 a RAISE
```

```
SQL state: P0001
```

### H.7.2) /\* INSERIMENTO PER TRIGGER DETTAGLIO PREZZO - QUANTITA \*/

INSERT INTO DETTAGLIO VALUES ('Fillo93','2022-02-12 ', '12:23:57 ',  
'Ipad2022ap ','199.99', ' 2');

#### Messages

```
ERROR: ERRORE:  IMPORTO INFERIORE AL DOVUTO  
CONTEXT:  funzione PL/pgSQL controlla_importo() riga 12 a RAISE
```

```
SQL state: P0001
```

### H.7.3) /\*INSERIMENTO PER TRIGGER PREZZO SCONTATO PROMOZIONE \*/

```
INSERT INTO PROMOZIONE VALUES ('Smart2gg2p', '00005', '16', 300);
```

#### Messages

```
ERROR: ERRORE:  il  prezzo scontato non può essere maggiore del
prezzo del prodotto
CONTEXT:  funzione PL/pgSQL controlla_offerta() riga 17 a RAISE
```

```
SQL state: P0001
```

### I) Inserimento dei dati

```
INSERT INTO SOCIO VALUES ( '0000000001', 'Fillo93');
INSERT INTO SOCIO VALUES ( '0000000002', 'Corradini34');
INSERT INTO SOCIO VALUES ( '0000000003', 'AndreaB93');
INSERT INTO SOCIO VALUES ( '0000000004', 'TeresaBar17');
INSERT INTO SOCIO VALUES ( '0000000005', 'Asiadelcu99');
INSERT INTO SOCIO VALUES ( '0000000006', 'Danigoldo');
INSERT INTO SOCIO VALUES ( '0000000007', 'Biunz94');
INSERT INTO SOCIO VALUES ( '0000000008', 'Marangoni');
INSERT INTO SOCIO VALUES ( '0000000009 ', 'DavideGorghetti');
INSERT INTO SOCIO VALUES ( '0000000010 ', 'Giuliaroccia');
INSERT INTO SOCIO VALUES ( '0000000011', 'AlbertoGuerzoni');
```

```
INSERT INTO CLIENTE VALUES ('Fillo93','Ciao2343', 'Modena',
'Canaletto' , '33', 'UDMEFR93A23F321T' , 'daodao@ciao.it', 'Filippo'
, 'Neri');
INSERT INTO CLIENTE VALUES ('Corradini34','Pescaazzurra', 'Milano',
'Navigli' , '15 ' , 'CCIDDN23B11O322E' , 'forzamilan@hotmail.it' , 'Davide'
, 'Corradini');
INSERT INTO CLIENTE VALUES ('AndreaB93 ', 'Sciranea', 'Roma',
'Tiberio', '94 ' , 'POASFA88E21F275T' , 'andreciao@libero.it', 'Andrea'
, 'Serafini');
INSERT INTO CLIENTE VALUES ('TeresaBar17','cgsgshoidgs',
'Modena', 'Campi', '5', 'CAPAFD90F30R234G' , 'musica@gmail.com',
'Teresa' , 'Picci ');
INSERT INTO CLIENTE VALUES ('Asiadelcu99','pisugshwd/()£&Q',
'Firenze', 'Medici', '9', 'PDFSAS99E30L542I ' , 'asiucciabella@msn.it',
'Asia' , 'Cucina');
INSERT INTO CLIENTE VALUES ('Danigoldo','Pilviaefsa', 'Napoli',
'Toledo', '324 ' , 'PDSAEH23P31D458V' , 'forzanapoli@aruba.it', 'Daniele'
, 'Remondini');
INSERT INTO CLIENTE VALUES ('Biunz94','Parigi2020', 'Roma',
'Cesare', '1', 'CCCDJH03E14R984U' , 'fabiunzello@gmail.it', 'Fabio'
, 'Luisi');
INSERT INTO CLIENTE VALUES ('Marangoni','Cavallobianco',
'Bologna', 'Indipendenza', '50', 'DOPAFS84R23I347D' ,
'bolosolobolo@internet.it', 'Mattia' , 'Vescovi');
```

INSERT INTO CLIENTE VALUES ('DavideGorghetti','ciaociaociao',  
'Modena', 'Albareto', '1', 'PDDSAM67E23D345F', 'nopubblicita@msn.it',  
'Davide', 'Gorghetti');

INSERT INTO CLIENTE VALUES ('AlbertoGuerzoni','Forzanapoli',  
'Napoli', 'San Biagio dei librai', '10', 'GRZLBR90L20A345E',  
'cavallinorampante@hotmail.it', 'Alberto', 'Guerzoni');

INSERT INTO PRODOTTO VALUES ('Ca750Refle','100', 'Fotocamera  
reflex', 'Canon 750D', 'Canon', 'Fotocamere', 399.95);

INSERT INTO PRODOTTO VALUES ('Ipad2022ap','30', 'Tablet Apple  
anno 2022 64 giga', 'Ipad', 'Apple', 'Tablet', 199.99);

INSERT INTO PRODOTTO VALUES ('Tvsams100p','20', 'Televisore  
Samsung 100 pollici', 'Black100', 'Samsung', 'Televisori', 299.99);

INSERT INTO PRODOTTO VALUES ('Nik1000ref','15', 'Fotocamera  
digitale', 'Nikon 1000', 'Nikon', 'Fotocamere', 99.99);

INSERT INTO PRODOTTO VALUES ('1231231233','15', 'Lavastoviglie  
di ultima generazione Wirpool', 'Wirpool 123', 'Wirpool',  
'Lavastoviglie', 100);

INSERT INTO PRODOTTO VALUES ('Gaming9999','30', 'Sedia da  
gaming', 'Gamingchair', 'Nusacc', 'Sedie da gaming', 150 );

INSERT INTO PRODOTTO VALUES ('Smart2gg2p','50', 'Smartbox per 2  
persone per 2 giorni', 'Smartbox', 'Smartbox', 'Smartbox', 99);

INSERT INTO PRODOTTO VALUES ('Flux40giri','40', 'Frullatore 40 giri  
al secondo imetech', 'Flux40', 'Imetech', 'Frullatore', 40);

INSERT INTO PRODOTTO VALUES ('Folletcasa','10', 'Pulitore  
autonomo Folletto', 'Folletto10', 'Folletto', 'Pulizia della casa', 200);

INSERT INTO PRODOTTO VALUES ('Bigice3000','50', 'Frigorifero di  
classe energetica A++', 'Bigice', 'Wirpool', 'Frigorifero', 750);

INSERT INTO PRODOTTO VALUES ('APEXLEGEND','50', 'Chiave di  
attivazione videogioco online', 'Apex\_legend', 'EA', 'Videogioco', 59.99);

INSERT INTO ORDINE VALUES('Fillo93','2022-02-12', '12:23:54',  
'Modena','Canaletto', '33');

INSERT INTO ORDINE VALUES('Fillo93','2019-04-30', '19:43:31',  
'Modena','Canaletto', '33');

INSERT INTO ORDINE VALUES('Danigoldo','2018-02-26', '15:32:34',  
'Napoli','Toledo', '324');

```

INSERT INTO ORDINE VALUES('Asiadelcu99','2021-06-30', '13:34:54',
'Firenze','Medici', '9');
INSERT INTO ORDINE VALUES('DavideGorghetti','2013-04-06',
'23:54:43', 'Modena ',' Albareto', '1');
INSERT INTO ORDINE VALUES('Marangoni','2015-07-05 ','22:53:31 ',
'Bologna ', 'Indipendenza' ,'50' );
INSERT INTO ORDINE VALUES('Marangoni','2016-02-22','21:05:08',
'Bologna ', 'Indipendenza ','50 ');
INSERT INTO ORDINE VALUES('TeresaBar17','2012-03-05', '10:15:00',
' Modena','Campi', '5');
INSERT INTO ORDINE VALUES('Biunz94','2022-11-13', '14:00:00',
'Roma ',' Cesare', '1 ');
INSERT INTO ORDINE VALUES('Biunz94','2022-12-15', '08:34:52',
'Roma ','Cesare ', '1 ');

```

```

INSERT INTO DETTAGLIO VALUES ('Fillo93','2022-02-12 ', '12:23:54 ',
'lpad2022ap ','199.99', ' 1');
INSERT INTO DETTAGLIO VALUES ('Fillo93','2019-04-30 ', '19:43:31 ',
'Folletcasa',' 400', '2');
INSERT INTO DETTAGLIO VALUES ('Danigoldo','2018-02-26 ',
'15:32:34 ', 'Bigice3000 ','750 ', '1 ');
INSERT INTO DETTAGLIO VALUES ('Asiadelcu99','2021-06-30 ', '
13:34:54', 'Bigice3000 ',' 750', ' 1');
INSERT INTO DETTAGLIO VALUES ('DavideGorghetti',' 2013-04-06',
'23:54:43', 'Smart2gg2p','198', '2');
INSERT INTO DETTAGLIO VALUES ('Marangoni','2015-07-05 ',
'22:53:31 ', 'Flux40giri','120', '3');
INSERT INTO DETTAGLIO VALUES ('Marangoni', '2016-02-22 ',
'21:05:08 ', 'Gaming9999','150', '1');
INSERT INTO DETTAGLIO VALUES ('TeresaBar17','2012-03-05 ',
'10:15:00 ', 'Tvsams100p','299.99', '1');
INSERT INTO DETTAGLIO VALUES ('Biunz94','2022-11-13 ', '14:00:00
', 'Nik1000ref','99.99', '1');
INSERT INTO DETTAGLIO VALUES ('Biunz94','2022-12-15 ', '08:34:52',
'Ca750Refle ', '399.95', '1');

```

```
INSERT INTO DETTAGLIO VALUES ('Asiadelcu99','2021-06-30 ', '
13:34:54', 'Gaming9999','150', '1');
INSERT INTO DETTAGLIO VALUES ('Biunz94','2022-12-15 ', '08:34:52',
'APEXLEGEND', '59.99', '1');
```

```
INSERT INTO RECENSIONE VALUES ('Fillo93','Ipad2022ap', '3',
'Ottimo prodotto ',' Questo tablet va che è una meraviglia');
INSERT INTO RECENSIONE VALUES ('Fillo93','Folletcasa', '4', 'casa
pulita ',' Arriva dove gli altri non arrivano');
INSERT INTO RECENSIONE VALUES ('Danigoldo','Bigice3000', '1',
'dimensioni ','le dimensioni non sono come quelle riportate ');
INSERT INTO RECENSIONE VALUES ('Asiadelcu99','Bigice3000', '2',
'consumi ',' bello, ma consuma troppo');
INSERT INTO RECENSIONE VALUES ('DavideGorghetti','Smart2gg2p',
'5', 'ovunque ',' ottima varietà di scelta');
INSERT INTO RECENSIONE VALUES ('Marangoni','Flux40giri', '2', '
rumore',' quando gira fa troppo rumore');
INSERT INTO RECENSIONE VALUES ('Marangoni','Gaming9999', '3',
'comoda',' Ottima sedia da gaming ');
INSERT INTO RECENSIONE VALUES ('TeresaBar17','Tvsams100p', '5',
'top ','colori perfetti');
INSERT INTO RECENSIONE VALUES ('Biunz94','Nik1000ref', '2', 'costo
',' costo troppo elevato per questo prodotto');
INSERT INTO RECENSIONE VALUES ('Biunz94','Ca750Refle', '1',
'ottiche ',' mancano le ottiche della stessa marca');
```

```
INSERT INTO PUNTO_VENDITA VALUES ('00001 ','32487548423', 'Via
dei pioppi 40 ', '059324232 ','store1@gmail.com ');
INSERT INTO PUNTO_VENDITA VALUES ('00002 ','23475762932', ' via
bella spesa 100', '327462322 ','store2@gmail.com');
INSERT INTO PUNTO_VENDITA VALUES ('00003 ','23905822732', 'Via
grande vista 23', '3264594232 ','store3@gmail.com');
INSERT INTO PUNTO_VENDITA VALUES ('00004','40487385873', 'Via
delle nazioni 99', '3925787545 ','store4@gmail.com');
INSERT INTO PUNTO_VENDITA VALUES ('00005','38457354463', 'via
del corso 3', '3827525433','store5@gmail.com');
```



```
INSERT INTO PROMOZIONE VALUES ('Smart2gg2p', '00001', '3',  
'49.99');  
INSERT INTO PROMOZIONE VALUES ('Smart2gg2p', '00004', '6',  
'49.99');  
INSERT INTO PROMOZIONE VALUES ('Flux40giri', '00005', '30', '20');  
INSERT INTO PROMOZIONE VALUES ('Ipad2022ap', '00002', '25',  
'100');  
INSERT INTO PROMOZIONE VALUES ('Bigice3000', '00004', '16',  
'599.99');  
INSERT INTO PROMOZIONE VALUES ('Tvsams100p', '00003', '17',  
'249.99');  
INSERT INTO PROMOZIONE VALUES ('Gaming9999', '00002', '18',  
'139.99');  
INSERT INTO PROMOZIONE VALUES ('Tvsams100p', '00001', '20',  
'199.99');  
INSERT INTO PROMOZIONE VALUES ('Smart2gg2p', '00005', '11',  
'69.99');  
INSERT INTO PROMOZIONE VALUES ('Bigice3000', '00005', '1',  
'69.99');
```

```
INSERT INTO FORNITORE VALUES  
( '23986745233','SDGRTG30T48F542E', 'Danithebst@gmail.com',  
'Danilo','Rossi','324523432');  
INSERT INTO FORNITORE VALUES  
( '23985742023','DGWPET23O10P457R', 'sandruccio35@libero.it',  
'Sandro ','Pertini ','345434245');  
INSERT INTO FORNITORE VALUES  
( '54243654321','CIOAEW30P05J458E', 'faccioridere@gmail.com', '  
Aldo','Baglio ', '3243257432');  
INSERT INTO FORNITORE VALUES  
( '42598547268','SDOISD20U03Y435E', 'brighella89@gmail.com', '  
Giacomo','Poretti ', '455432345');  
INSERT INTO FORNITORE VALUES  
( '32598682211','PPDSAT04P04G342T', 'alfosigno@gmail.com', '  
Alfonso',' Signorini','32343456');
```

```
INSERT INTO FORNITURA VALUES ('23986745233', '2022-03-03');
INSERT INTO FORNITURA VALUES ('23985742023', '2021-04-04');
INSERT INTO FORNITURA VALUES ('23986745233', '2019-06-30');
INSERT INTO FORNITURA VALUES ('23985742023', '2017-01-03');
INSERT INTO FORNITURA VALUES ('23986745233', '2015-03-07');
INSERT INTO FORNITURA VALUES ('54243654321', '2016-08-17');
INSERT INTO FORNITURA VALUES ('42598547268', '2014-05-19');
INSERT INTO FORNITURA VALUES ('42598547268', '2020-12-14');
INSERT INTO FORNITURA VALUES ('32598682211', '2021-09-09');
INSERT INTO FORNITURA VALUES ('32598682211', '2012-12-21');
```

```
INSERT INTO DI VALUES ('23986745233','2022-03-03', 'Flux40giri',
'23');
INSERT INTO DI VALUES ('23985742023','2021-04-04', 'Gaming9999',
'4');
INSERT INTO DI VALUES ('23986745233','2019-06-30', 'Smart2gg2p',
'52');
INSERT INTO DI VALUES ('23985742023','2017-01-03', 'Nik1000ref',
'14');
INSERT INTO DI VALUES ('23986745233', '2015-03-07' , '1231231233',
'5');
INSERT INTO DI VALUES ('54243654321','2016-08-17', 'Ipad2022ap',
'16');
INSERT INTO DI VALUES ('42598547268','2014-05-19', 'Folletcasa', '3');
INSERT INTO DI VALUES ('42598547268','2020-12-14', 'Bigice3000',
'7');
INSERT INTO DI VALUES ('32598682211','2021-09-09', 'Ca750Refle',
'11');
INSERT INTO DI VALUES ('32598682211','2012-12-21', 'Tvsams100p',
'3');
```

```
INSERT INTO PRESENTE_IN VALUES ('00001', 'Tvsams100p');
INSERT INTO PRESENTE_IN VALUES ('00001', 'Ca750Refle');
INSERT INTO PRESENTE_IN VALUES ('00005', 'Bigice3000');
INSERT INTO PRESENTE_IN VALUES ('00003', 'Bigice3000');
INSERT INTO PRESENTE_IN VALUES ('00001', 'Folletcasa');
INSERT INTO PRESENTE_IN VALUES ('00003', 'Ipad2022ap');
```

```
INSERT INTO PRESENTE_IN VALUES ('00002', '1231231233');
INSERT INTO PRESENTE_IN VALUES ('00002', 'Smart2gg2p');
INSERT INTO PRESENTE_IN VALUES ('00004', 'Nik1000ref');
INSERT INTO PRESENTE_IN VALUES ('00004', 'Folletcasa');
```

```
INSERT INTO ACQUISTA VALUES ('00000000004', '00004',
'Bigice3000');
INSERT INTO ACQUISTA VALUES ('00000000006', '00002',
'Ipad2022ap');
INSERT INTO ACQUISTA VALUES ('00000000008', '00003',
'Tvsams100p');
INSERT INTO ACQUISTA VALUES ('00000000001', '00005',
'Smart2gg2p');
INSERT INTO ACQUISTA VALUES ('00000000003', '00002',
'Gaming9999');
INSERT INTO ACQUISTA VALUES ('00000000002', '00003',
'Ipad2022ap');
INSERT INTO ACQUISTA VALUES ('00000000007', '00001',
'Ca750Refle');
```

```
INSERT INTO CITTA VALUES ('Modena');
INSERT INTO CITTA VALUES ('Bologna');
INSERT INTO CITTA VALUES ('Firenze');
INSERT INTO CITTA VALUES ('Roma');
INSERT INTO CITTA VALUES ('Napoli');
```

```
INSERT INTO PRESENTE VALUES ('Modena', '00001');
INSERT INTO PRESENTE VALUES ('Bologna', '00002');
INSERT INTO PRESENTE VALUES ('Firenze', '00003');
INSERT INTO PRESENTE VALUES ('Roma', '00004');
INSERT INTO PRESENTE VALUES ('Napoli', '00005');
```

```
INSERT INTO RESPONSABILE VALUES ('000001',
'SPDFSA12P03T432P ', 'Alfredi@gmail.com ', 'Alfredo ', 'Uccello ',
'000002', '3432345434');
```

INSERT INTO RESPONSABILE VALUES ('000002',  
'DSASDF10R06R988W ',' Willesp@gmail.com', 'William ','Esposito  
,','000001','234565434');

INSERT INTO RESPONSABILE VALUES ('000003',  
'PDFRTV30R12P456E','Lucaran@gmail.com ', 'Luca ',' Ranieri',  
'000001','05973382');

INSERT INTO RESPONSABILE VALUES ('000004',  
'GTRFBC09U12H543S',' Robbivalle@gmail.com', 'Roberto ',' Valle',  
'000005','432343432');

INSERT INTO RESPONSABILE VALUES ('000005',  
'LMNSDR07I19U435E','Silvietta@gmail.com ', ' Silvia','Pellacani ',  
'000004', '3456543456');

INSERT INTO DIPENDENTE VALUES ('100000', 'Addetto casse',  
'PLDQRT35G24H876F','Lupolucio@gmail.com ', 'Lucio ', 'Simonini  
,','23434434');

INSERT INTO DIPENDENTE VALUES ('100001', 'Costumer service',  
'BVRTCV08R04P386G','Lucilot@gmail.com ', 'Lucia ', 'Lotti  
,','34345654');

INSERT INTO DIPENDENTE VALUES ('100002', 'Ordinamento scaffali ',  
'CARVAD21I05V346V','Quartoequinto@gmail.com ', 'Quarto ', ' Rossi'  
,','4543454323');

INSERT INTO DIPENDENTE VALUES ('100003', 'Pulizia ',  
'VBTRMG09O31I653E','Valepep@gmail.com ', ' Valerio' , 'Peppini  
,','3234565456' );

INSERT INTO DIPENDENTE VALUES ('100004', 'Guardia giurata ',  
'CCCGNN97M18F257T','Carletta@gmail.com ', 'Carla ', 'Grandi  
,','687546327');

INSERT INTO DIPENDENTE VALUES ('100005', 'Fattorino',  
'BLDBTR24P52S123S','Aliventu@gmail.com', ' Alice' , '  
Venturi','345834956');

INSERT INTO DIPENDENTE VALUES ('100006', 'Cassiere',  
'CTVREC04T05T928Z','Alleselmi@gmail.com ', 'Alessia ', 'Selmi ',  
,','3748393542');

INSERT INTO DIPENDENTE VALUES ('100007', 'Sindacalista',  
'NBVRTC09E05P243T','Milbarzo@gmail.com', 'Milena ', 'Barozzi  
,','3425843586');

```
INSERT INTO DIPENDENTE VALUES ('100008', 'Infermiere',  
'MPRVET04P06M542C', 'Simoall@gmail.com', 'Simone', '  
Allegri', '47582369');  
INSERT INTO DIPENDENTE VALUES ('100009', 'Tecnico informatico',  
'CRVTPC11L10C325T', 'Maurisan@gmail.com', 'Maurizio ', '  
Santini', '245235654' );
```

```
INSERT INTO RESPONSABILE_DI VALUES ('000001', '00001');  
INSERT INTO RESPONSABILE_DI VALUES ('000002', '00002');  
INSERT INTO RESPONSABILE_DI VALUES ('000003', '00003 ');  
INSERT INTO RESPONSABILE_DI VALUES ('000004', '00004');  
INSERT INTO RESPONSABILE_DI VALUES ('000005', '00005');
```

```
INSERT INTO LAVORA VALUES ('100000', '00001');  
INSERT INTO LAVORA VALUES ('100001', '00002');  
INSERT INTO LAVORA VALUES ('100002', '00002');  
INSERT INTO LAVORA VALUES ('100003', '00003');  
INSERT INTO LAVORA VALUES ('100004', '00004');  
INSERT INTO LAVORA VALUES ('100005', '00005');  
INSERT INTO LAVORA VALUES ('100006', '00005');  
INSERT INTO LAVORA VALUES ('100007', '00004');  
INSERT INTO LAVORA VALUES ('100008', '00002');  
INSERT INTO LAVORA VALUES ('100009', '00001');
```

## J) Operazioni in SQL

### J.1) Query di aggiornamento

#### J.1.1) UPDATE PROMOZIONE SET PREZZO\_SCONTATO=15 WHERE CODICE\_PV='00001'

```
1 select *
2 from promozione
3 where codice_pv='00001'
```

	codice_prodotto [PK] character (10)	codice_pv [PK] character (5)	settimana integer	prezzo_scontato real
1	Smart2gg2p	00001	3	49.99
2	Tvsams100p	00001	20	199.99

```
1 UPDATE PROMOZIONE SET PREZZO_SCONTATO=15
2 WHERE CODICE_PV='00001'
```

	codice_prodotto [PK] character (10)	codice_pv [PK] character (5)	settimana integer	prezzo_scontato real
1	Smart2gg2p	00001	3	15
2	Tvsams100p	00001	20	15

#### J.1.2) UPDATE DIPENDENTE SET EMAIL='Quartorossi@gmail.com' WHERE CODICE\_DIPENDENTE='100003'

```
1 SELECT *
2 FROM DIPENDENTE
3 WHERE MATRICOLA_DIPENDENTE='100003'
```

	matricola_dipendente [PK] character (6)	qualifica character varying (100)	cf character (16)	email character varying (40)
1	100003	Pulizia	VBTRMG09031I653E	Valepep@gmail.com

```
1 UPDATE DIPENDENTE
2 SET EMAIL='Quartorossi@gmail.com'
3 WHERE MATRICOLA_DIPENDENTE='100003'
```

	matricola_dipendente [PK] character (6)	qualifica character varying (100)	cf character (16)	email character varying (40)
1	100003	Pulizia	VBTRMG09031I653E	Quartorossi@gmail.com

### J.1.3) UPDATE RECENSIONE SET STELLE=5 WHERE CODICE\_PRODOTTO= 'Bigice3000' AND USERNAME='Danigoldo'

```
1 select *
2 from recensione
3 where username='Danigoldo'
4
```

Data Output	Explain	Messages	Notifications
username	codice_prodotto	stelle	
[PK] character varying (100)	[PK] character (10)	smallint	
1 Danigoldo	Bigice3000	1	

Query Editor Query History

```
1 UPDATE RECENSIONE SET STELLE=5
2 WHERE CODICE_PRODOTTO= 'Bigice3000'
3 AND USERNAME='Danigoldo'
```

Data Output	Explain	Messages	Notifications
username	codice_prodotto	stelle	
[PK] character varying (100)	[PK] character (10)	smallint	
1 Danigoldo	Bigice3000	5	

## J.2) Query di eliminazione

### J.2.1)DELETE FROM SOCIO WHERE COD\_TESSERA= '0000000003'

```
1 select *
2 from socio
```

Data Output	Explain	Messages	Notifications
cod_tessera	username		
[PK] character (10)	character varying (100)		
1 0000000001	Fillo93		
2 0000000002	Corradini34		
3 0000000003	AndreaB93		
4 0000000004	TeresaBar17		
5 0000000005	Asiadelcu99		
6 0000000006	Danigoldo		
7 0000000007	Biunz94		
8 0000000008	Marangoni		
9 0000000009	DavideGorghetti		
10 0000000010	Giuliaroccia		
11 0000000011	AlbertoGuerzoni		

```
1 DELETE FROM SOCIO WHERE COD_TESSERA= '0000000003'
```

Data Output	Explain	Messages	Notifications
cod_tessera	username		
[PK] character (10)	character varying (100)		
1 0000000001	Fillo93		
2 0000000002	Corradini34		
3 0000000004	TeresaBar17		
4 0000000005	Asiadelcu99		
5 0000000006	Danigoldo		
6 0000000007	Biunz94		
7 0000000008	Marangoni		
8 0000000009	DavideGorghetti		
9 0000000010	Giuliaroccia		
10 0000000011	AlbertoGuerzoni		

## J.2.2) DELETE FROM FORNITORE WHERE PIVAF='23986745233'

```
1 SELECT *
2 FROM FORNITORE
3
```

	Data Output	Explain	Messages	Notifications
	<b>pivaf</b> [PK] character (11)	<b>cf</b> character (16)	<b>email</b> character varying (40)	<b>nome</b> character varying (100)
1	23986745233	SDGRTG30T48F542E	Danithebst@gmail.com	Danilo
2	23985742023	DGWPET23O10P457R	sandrucchio35@libero.it	Sandro
3	54243654321	CIOAEW30P05J458E	faccioridere@gmail.com	[...] Aldo
4	42598547268	SDOISD20U03Y435E	brighella89@gmail.com	[...] Giacomo
5	32598682211	PPDSAT04P04G342T	alfosigno@gmail.com	[...] Alfonso

```
1 DELETE FROM FORNITORE WHERE PIVAF='23986745233'
```

	Data Output	Explain	Messages	Notifications
	<b>pivaf</b> [PK] character (11)	<b>cf</b> character (16)	<b>email</b> character varying (40)	
1	23985742023	DGWPET23O10P457R	sandrucchio35@libero.it	
2	54243654321	CIOAEW30P05J458E	faccioridere@gmail.com	
3	42598547268	SDOISD20U03Y435E	brighella89@gmail.com	
4	32598682211	PPDSAT04P04G342T	alfosigno@gmail.com	

## J.2.3) DELETE FROM RESPONSABILE WHERE MATRICOLA\_RESPONSABILE='000003'

```
1 SELECT * FROM RESPONSABILE
```

	Data Output	Explain	Messages	Notifications
	<b>matricola_responsabile</b> [PK] character (6)	<b>cf</b> character (16)	<b>email</b> character varying (40)	
1	000001	SPDFSA12P03T432P	Alfredi@gmail.com	
2	000002	DSASDF10R06R988W	[...] Willesp@gmail.com	
3	000003	PDFRTV30R12P456E	Lucaran@gmail.com	
4	000004	GTRFBC09U12H543S	[...] Robbivalle@gmail.com	
5	000005	LMNSDR07I19U435E	Silvietta@gmail.com	

```
1 DELETE FROM RESPONSABILE WHERE MATRICOLA_RESPONSABILE='000003'
```

	Data Output	Explain	Messages	Notifications
	<b>matricola_responsabile</b> [PK] character (6)	<b>cf</b> character (16)	<b>email</b> character varying (40)	<b>nome</b> character varying
1	000001	SPDFSA12P03T432P	Alfredi@gmail.com	Alfredo
2	000002	DSASDF10R06R988W	[...] Willesp@gmail.com	William
3	000004	GTRFBC09U12H543S	[...] Robbivalle@gmail.com	Roberto
4	000005	LMNSDR07I19U435E	Silvietta@gmail.com	[...] Silvia



## H) INTERROGAZIONI

Qui vengono riportate solo alcune delle possibili interrogazioni

### H.1) Selezionare i prodotti ordinati dal cliente “nomecliente”

```
SELECT *  
FROM PRODOTTO P, DETTAGLIO D  
WHERE P.CODICE_PRODOTTO = D.CODICE_PRODOTTO  
AND D.USERNAME = 'nomecliente'
```

Query Editor

Query History

1

SELECT \*

2

FROM PRODOTTO P, DETTAGLIO D

3

WHERE P.CODICE\_PRODOTTO = D.CODICE\_PRODOTTO

4

AND D.USERNAME = 'Fillo93'

5

Data Output

Explain

Messages

Notifications

<div><div><div></div></div></div> <div>codice_prodotto</div> <div>character (10)</div> <div><div><div></div></div></div>
--

 quantita\_residua  integer |

 descrizione  character varying (200) | nome  character varying (100) | marca  character varying (100) | categoria  character varying (50) | prezzo  real | username  character varying (100) | data  date || 1 | Folletcasa | 10 | Pulitore autonomo Folletto | Folletto10 | Folletto | Pulizia della casa | 200 | Fillo93 | 2019-04-25 |
| 2 | Ipad2022ap | 30 | Tablet Apple anno 2022 64 giga | Ipad | Apple | Tablet | 199.99 | Fillo93 | 2022-02-15 |

### H.2) Selezionare i prodotti che NON sono stati ordinati dal cliente “nomecliente”

```
SELECT *  
FROM PRODOTTO P  
WHERE P.CODICE_PRODOTTO NOT IN (  
    SELECT CODICE_PRODOTTO  
    FROM DETTAGLIO D  
    WHERE D.USERNAME = 'nomecliente' )
```

```

1 SELECT *
2 FROM PRODOTTO P
3 WHERE P.CODICE_PRODOTTO NOT IN (
4     SELECT CODICE_PRODOTTO
5     FROM DETTAGLIO D
6     WHERE D.USERNAME = 'Marangoni' )

```

Data Output Explain Messages Notifications

	codice_prodotto [PK] character (10)	quantita_residua integer	descrizione character varying (200)	nome character varying (100)	marca character varying (100)	categoria character varying (50)	prezzo real
1	Ca750Refle	100	Fotocamera reflex	[...] Canon 750D	Canon	Fotocamere	399.95
2	Ipad2022ap	30	Tablet Apple anno 2022 64 giga	Ipad	Apple	Tablet	199.99
3	Tvsams100p	20	Televisore Samsung 100 pollici	Black100	Samsung	Televisori	299.99
4	Nik1000ref	15	Fotocamera digitale	Nikon 1000	Nikon	Fotocamere	99.99
5	1231231233	15	Lavastoviglie di ultima generazione Wirpool	Wirpool 123	Wirpool	Lavastoviglie	100
6	Smart2gg2p	50	Smartbox per 2 persone per 2 giorni	Smartbox	[...] Smartbox	Smartbox	99
7	Folletcasa	10	Pulitore autonomo Folletto	Folletto10	Folletto	Pulizia della casa	200
8	Bigice3000	50	Frigorifero di classe energetica A++	Bigice	Wirpool	Frigorifero	750
9	APEXLEGEND	50	Chiave di attivazione videogiochi online	Apex_legend	EA	Videogioco	59.99

### H.3) Selezionare i clienti che hanno ordinato un prodotto “Sedia da gaming” E “Frigorifero di classe energetica A++”

SELECT USERNAME

FROM CLIENTE

WHERE USERNAME IN

((SELECT USERNAME

FROM DETTAGLIO D, PRODOTTO P

WHERE D.CODICE\_PRODOTTO = P.CODICE\_PRODOTTO

AND P.DESCRIZIONE = ‘Sedia da gaming ’)

INTERSECT

(SELECT USERNAME

FROM DETTAGLIO D, PRODOTTO P

WHERE D.CODICE\_PRODOTTO = P.CODICE\_PRODOTTO

AND P.DESCRIZIONE = ‘Frigorifero di classe energetica A++’))

```

1  SELECT USERNAME
2  FROM CLIENTE
3  WHERE USERNAME IN
4      ((SELECT USERNAME
5         FROM DETTAGLIO D, PRODOTTO P
6         WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
7         AND P.DESCRIZIONE = 'Sedia da gaming ')
8  INTERSECT
9      (SELECT USERNAME
10     FROM DETTAGLIO D, PRODOTTO P
11     WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
12     AND P.DESCRIZIONE = 'Frigorifero di classe energetica A++'))
13

```

Data Output Explain Messages Notifications

	username [PK] character varying (100)	
1	Asiadelcu99	

#### H.4) Selezionare i clienti che hanno ordinato un prodotto “Televisore Samsung 100 pollici” O “Smartbox per 2 persone per 2 giorni”

```

SELECT *
FROM CLIENTE
WHERE USERNAME IN
    ((SELECT USERNAME
       FROM DETTAGLIO D, PRODOTTO P
       WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
       AND P.DESCRIZIONE = 'Televisore Samsung 100 pollici')
    UNION
    (SELECT USERNAME
     FROM DETTAGLIO D, PRODOTTO P
     WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
     AND P.DESCRIZIONE = 'Smartbox per 2 persone per 2 giorni'))

```

1	SELECT *
2	FROM CLIENTE
3	WHERE USERNAME IN
4	((SELECT USERNAME
5	FROM DETTAGLIO D, PRODOTTO P
6	WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
7	AND P.DESCRIZIONE = 'Televisore Samsung 100 pollici')
8	UNION
9	(SELECT USERNAME
10	FROM DETTAGLIO D, PRODOTTO P
11	WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
12	AND P.DESCRIZIONE = 'Smartbox per 2 persone per 2 giorni'))
13	

Data Output	Explain	Messages	Notifications
-------------	---------	----------	---------------

	username [PK] character varying (100)	password character varying (25)	citta character varying (50)	via character varying (100)
1	TeresaBar17	csgshoidgs	Modena	Campi
2	DavideGorghetti	ciaociaociao	Modena	Albareto

## H.5) Estrarre le città i quali punti vendita hanno fatto una promozione nella prima settimana dell'anno

```
SELECT P.NOME
FROM PRESENTE P, PUNTO_VENDITA PV, PROMOZIONE PR
WHERE P.CODICE_PV=PV.CODICE_PV AND
PV.CODICE_PV=PR.CODICE_PV AND PR.SETTIMANA =1
```

1	SELECT P.NOME
2	FROM PRESENTE P, PUNTO_VENDITA PV, PROMOZIONE PR
3	WHERE P.CODICE_PV=PV.CODICE_PV AND PV.CODICE_PV=PR.CODICE_PV AND Pr.SETTIMANA =1
4	

Data Output	Explain	Messages	Notifications
-------------	---------	----------	---------------

	nome character varying (100)
1	Napoli

## H.6) Selezionare il fornitore che ha fornito più prodotti

```
SELECT DI.pivaf, F.nome
FROM FORNITORE F, DI
WHERE DI.PIVAF=F.PIVAF
```

```

GROUP BY DI.PIVAF, F.nome
HAVING SUM(QUANTITA) >= ALL
    (SELECT SUM(QUANTITA)
     FROM FORNITORE F, DI
     WHERE DI.PIVAF=F.PIVAF
     GROUP BY DI.PIVAF)



```

```

1  SELECT DI.pivaf, F.nome
2  FROM FORNITORE F, DI
3  WHERE DI.PIVAF=F.PIVAF
4  GROUP BY DI.PIVAF, F.nome
5  HAVING SUM(QUANTITA) >= ALL
6      (SELECT SUM(QUANTITA)
7       FROM FORNITORE F, DI
8       WHERE DI.PIVAF=F.PIVAF
9       GROUP BY DI.PIVAF)
10

```

Data Output   Explain   Messages   Notifications

	pivaf character (11) 	nome character varying (100) 	
1	23986745233	Danilo	

**H.7) Per ogni punto vendita calcolare il fatturato ed ordinarlo in ordine decrescente**

```

SELECT CODICE_PV, SUM(D.IMPORTO*D.QUANTITA) AS
FATTURATO
FROM PRESENTE_IN P, DETTAGLIO D
WHERE P.CODICE_PRODOTTO=D.CODICE_PRODOTTO
GROUP BY CODICE_PV
ORDER BY 2 DESC



```

```

1 SELECT CODICE_PV, SUM(D.IMPORTO*D.QUANTITA) AS FATTURATO
2 FROM PRESENTE_IN P, DETTAGLIO D
3 WHERE P.CODICE_PRODOTTO=D.CODICE_PRODOTTO
4 GROUP BY CODICE_PV
5 ORDER BY 2 DESC

```

Data Output Explain Messages Notifications

	<b>codice_pv</b> character (5) 	<b>fatturato</b> double precision 	
1	00003	1699.990005493164	
2	00005	1500	
3	00001	1499.9400024414062	
4	00004	899.9899978637695	
5	00002	396	

**H.8)Selezionare il punto vendita nel quale lavora il maggior numero di dipendenti**

```

SELECT P.CODICE_PV
FROM PUNTO_VENDITA P, LAVORA L
WHERE P.CODICE_PV = L.CODICE_PV
GROUP BY P.CODICE_PV
HAVING COUNT (*) >= ALL (
    SELECT COUNT (*)
    FROM PUNTO_VENDITA P1, LAVORA L1
    WHERE P1.CODICE_PV = L1.CODICE_PV
    GROUP BY P1.CODICE_PV)

```

```

1 SELECT P.CODICE_PV
2 FROM PUNTO_VENDITA P, LAVORA L
3 WHERE P.CODICE_PV = L.CODICE_PV
4 GROUP BY P.CODICE_PV
5 HAVING COUNT (*) >= ALL (
6     SELECT COUNT (*)
7     FROM PUNTO_VENDITA P1, LAVORA L1
8     WHERE P1.CODICE_PV = L1.CODICE_PV
9     GROUP BY P1.CODICE_PV)

```

Data Output Explain Messages Notifications

	codice_pv [PK] character (5) 	
1	00002	

### H.9) Clienti che hanno ordinato tutti i prodotti di categoria “videogiochi”

=> Clienti per i quali non esiste un prodotto di categoria “videogioco” che essi NON hanno ordinato

```

SELECT *
FROM CLIENTE C
WHERE NOT EXISTS (
    SELECT *
    FROM PRODOTTO P
    WHERE P.CATEGORIA = 'Videogioco'
    AND NOT EXISTS(
        SELECT *
        FROM
        DETTAGLIO D
        WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
        AND D.USERNAME = C.USERNAME))





```

```

1  SELECT *
2  FROM CLIENTE C
3  WHERE NOT EXISTS (
4      SELECT *
5      FROM PRODOTTO P
6      WHERE P.CATEGORIA = 'Videogioco'
7      AND NOT EXISTS(
8          SELECT *
9          FROM
10             DETTAGLIO D
11             WHERE D.CODICE_PRODOTTO = P.CODICE_PRODOTTO
12             AND D.USERNAME = C.USERNAME))

```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

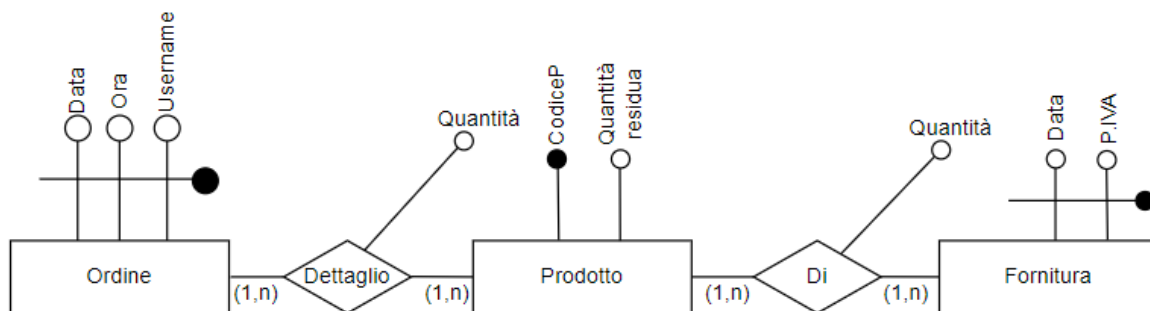
	username [PK] character varying (100) 	password character varying (25) 	citta character varying (50) 	via character varying (100) 
1	Biunz94	Parigi2020	Roma	Cesare



## K) Studio di un dato derivato

Il dato derivato in questione è la **quantità residua** di ogni singolo prodotto, che può essere memorizzata come attributo dell'entità prodotto, oppure può essere ottenuta come somma di tutte le forniture del prodotto in questione, meno ogni ordine del prodotto in questione.

Per valutare la convenienza o meno del mantenimento del dato derivato supponiamo trascurabile l'occupazione di memoria di quest'ultimo, ed andiamo invece a considerare il numero di accessi giornalieri che vengono effettuati per tenere aggiornata la quantità residua oppure per visualizzare ogni attributo del prodotto (inclusa la quantità residua).



Le operazioni prese in considerazione sono:

- 1) Inserimento di un nuovo ordine di un prodotto, supposto valido il codice del prodotto.
- 2) Inserimento di una nuova fornitura di un prodotto, supposto valido il codice del prodotto.
- 3) Visualizzazione di tutti i dati di un prodotto (inclusa la quantità residua).

Per semplicità si considera la fornitura di un prodotto e l'ordine di un prodotto, anche se sia nella fornitura, sia nell'ordine possono essere presenti più prodotti diversi

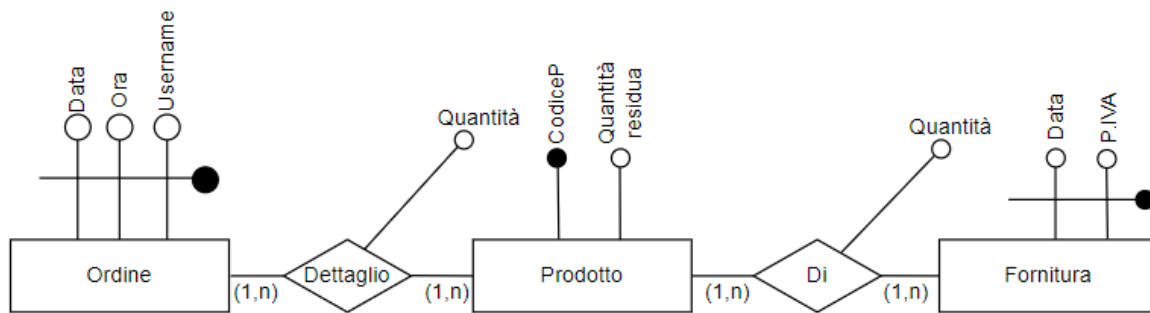
**Tabella dei volumi**

Concetto	Tipo	Volume Dati
Ordine	E	10 000
Prodotto	E	200
Fornitura	E	5 000
Dettaglio	A	30 000
Di	A	20 000

**Tabella delle operazioni**

Operazione	Tipo	Frequenza
Operazione 1	I	20/gg
Operazione 2	I	1/gg
Operazione 3	I	5/gg

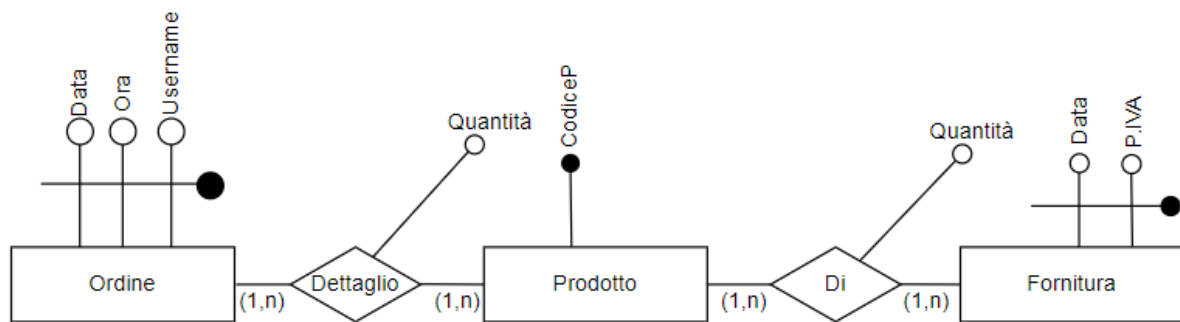
## K.1) Con dato derivato



In questo caso lo schema rimane il medesimo

Concetto	Accesso	Tipo	Totale
Ordine	1	S	
Dettaglio	1	S	
Prodotto	1	L	
Prodotto	1	S	
<b>OP1: 7 * 20</b>			<b>140 acc/giorno</b>
Fornitura	1	S	
Di	1	S	
Prodotto	1	L	
Prodotto	1	S	
<b>OP2: 7 * 1</b>			<b>7 acc/giorno</b>
Prodotto	1	L	
<b>OP3: 1 * 5</b>			<b>5 acc/giorno</b>
<b>Totale</b>			<b>152 acc/giorno</b>

## K.2) Senza dato derivato



Concetto	Accesso	Tipo	Totale
Ordine	1	S	
Dettaglio	1	S	
<b>OP1: 4*20</b>			<b>80 acc/giorno</b>
Fornitura	1	S	
Di	1	S	
<b>OP2: 4*1</b>			<b>4 acc/giorno</b>
Prodotto	1	L	
Dettaglio	150	L	
Fornitura	100	L	
<b>OP3: 251 * 5</b>			<b>1255 acc/giorno</b>
<b>Totale</b>			<b>1339 acc/giorno</b>

Dettaglio/Prodotto =  $30000 / 200 = 150$  : Un prodotto è contenuto in media in 150 dettagli diversi

Di/Prodotto =  $20000 / 200 = 100$  : Un prodotto è contenuto in media in 100 forniture diverse

In conclusione è conveniente **mantenere** il dato derivato quantità residua.

## L) PROGETTO FISICO

Data la query

```
SELECT *  
FROM RECENSIONI R, PRODOTTO P  
WHERE R.CODPROD = P.CODPROD  
AND R.STELLE >= 3  
AND P.CATEGORIA = 'Computer'  
AND P.MARCA IN ('Apple', 'Asus')
```

Le relazioni sono

PRODOTTO (CODPROD, MARCA, MODELLO, NOME, CATEGORIA...)  
RECENSIONE (CODPROD, USERNAME, STELLE, TITOLO, TESTO)

**FK: CODPROD REFERENCES** PRODOTTO

**FK: USERNAME REFERENCES** CLIENTE

Supponendo che PRODOTTI contenga 10.000 tuple (NT) in 1.000 pagine (NB), mentre RECENSIONI contenga 500 tuple (NT) in 50 pagine (NB).

Gli indici sono tutti di tipo unclustered e sono

- a) Marca con 50 valori distinti (NK) in 10 foglie (NF)
- b) Stelle con 9 valori distinti (NK) in 5 foglie (NF) valori di stelle da 1 a 5
- c) Categoria con 15 valori distinti (NK) in 5 foglie (NF)
- d) R.Codprodotto con 100 valori distinti (NK) in 10 (NF)
- e) P.Codprodotto con 100 valori distinti (NK) in 15 (NF)

Recensione  $\Rightarrow$  Prodotto

Recensione)

Cseq= 50

F stella =  $(5-3)/(5-1) = \frac{1}{2}$

C stella =  $\text{Int sup}((\frac{1}{2} * \text{NF}) + (\frac{1}{2} * \text{NT})) \Rightarrow (\frac{1}{2} * 5) + (\frac{1}{2} * 500) = 3 + 250 = 253$

E =  $\text{Int sup}(\text{NT} * \frac{1}{2}) = 500 * \frac{1}{2} = 250$

Prodotto)

$$C_{seq} = 1000$$

$$F_{categoria} = 1/NK = 1/15$$

$$F_{marca} = n/NK = 2/50 = 1/25$$

$$C_{categoria} = \text{Int sup} ( 1/15 * NF ) + ( 1/15 * NT ) = (1/15*5) + (1/15*10000) = 1+667 = 668$$

$$C_{marca} = \text{Int sup} ( 1/25 * NF ) + ( 1/25 * NT ) = (1/25*10) + (1/25*10000) = 1+400 = 401$$

$$F_{Codprodotto} = 1/100$$

$$C_{Codprodotto} = \text{Int sup} (1/100 * NF) + (1/100 * NT) = (1/100*15) + (1/100*10000) = 1+100 = 101$$

$$\begin{aligned} C_{join} &= C_{recensione} + E_{recensione} * C_{prodotto} \\ &= 50 + 250*101 \\ &= 50 + 25250 = 25300 \end{aligned}$$

Prodotto  $\Rightarrow$  Recensione

Prodotto)

$$C_{seq} = 1000$$

$$F_{categoria} = 1/15$$

$$F_{marca} = 1/25$$

$$C_{categoria} = 668$$

$$C_{marca} = 401$$

$$E = \text{Int sup} (NT * F_{categoria} * F_{marca}) = 10000*1/15*1/25 = 27$$

Recensione)

$$C_{seq} = 50$$

$$F_{stella} = 1/2$$

$$C_{stella} = 253$$

$$F_{codprodotto} = 1/100$$

$$C_{codprodotto} = \text{Int sup} ( 1/100 * NF ) + ( 1/100 * NT ) = (1/100*10) + (1/100*500) = 1+5 = 6$$

$$\begin{aligned} C_{join} &= C_{prodotto} + E_{prodotto} * C_{recensione} \\ &= 401 + 27*6 = 401 + 162 = 563 \end{aligned}$$

$$563 < 25300$$

è preferibile la sequenza Prodotto  $\Rightarrow$  Recensione

## M) JAVA JDBC

JDBC ci consente di utilizzare il database all'interno di un programma java. Sono state create le classi Login, Connections, e Menu.

Nella classe Login viene richiesto all'utente di inserire il proprio username e la propria password per connettersi al DB, nella classe Connections viene effettuata la connessione con le credenziali fornite, se la connessione va a buon fine viene creata un'istanza della classe Menu. Nella classe Menu, viene visualizzato un piccolo menu con qualche operazione basilare come inserimenti ed interrogazioni semplici. Il tutto viene implementato via command line interface.

```
1 per registrare un nuovo cliente
2 per inserire un nuovo prodotto
3 per inserire una nuova recensione
4 per visualizzare tutti gli ordini di un cliente
5 per visualizzare tutte le recensioni di un prodotto
1997 per terminare
```