

**Laboratorio**  
**Corso di Basi di Dati e Web**  
**A.A. 2019/2020**

**Progetto “Catena\_di\_supermercati”**

**Gruppo composto da:**

*Giulia Scarpa giulia.scarpa3@studenti.unimi.it*

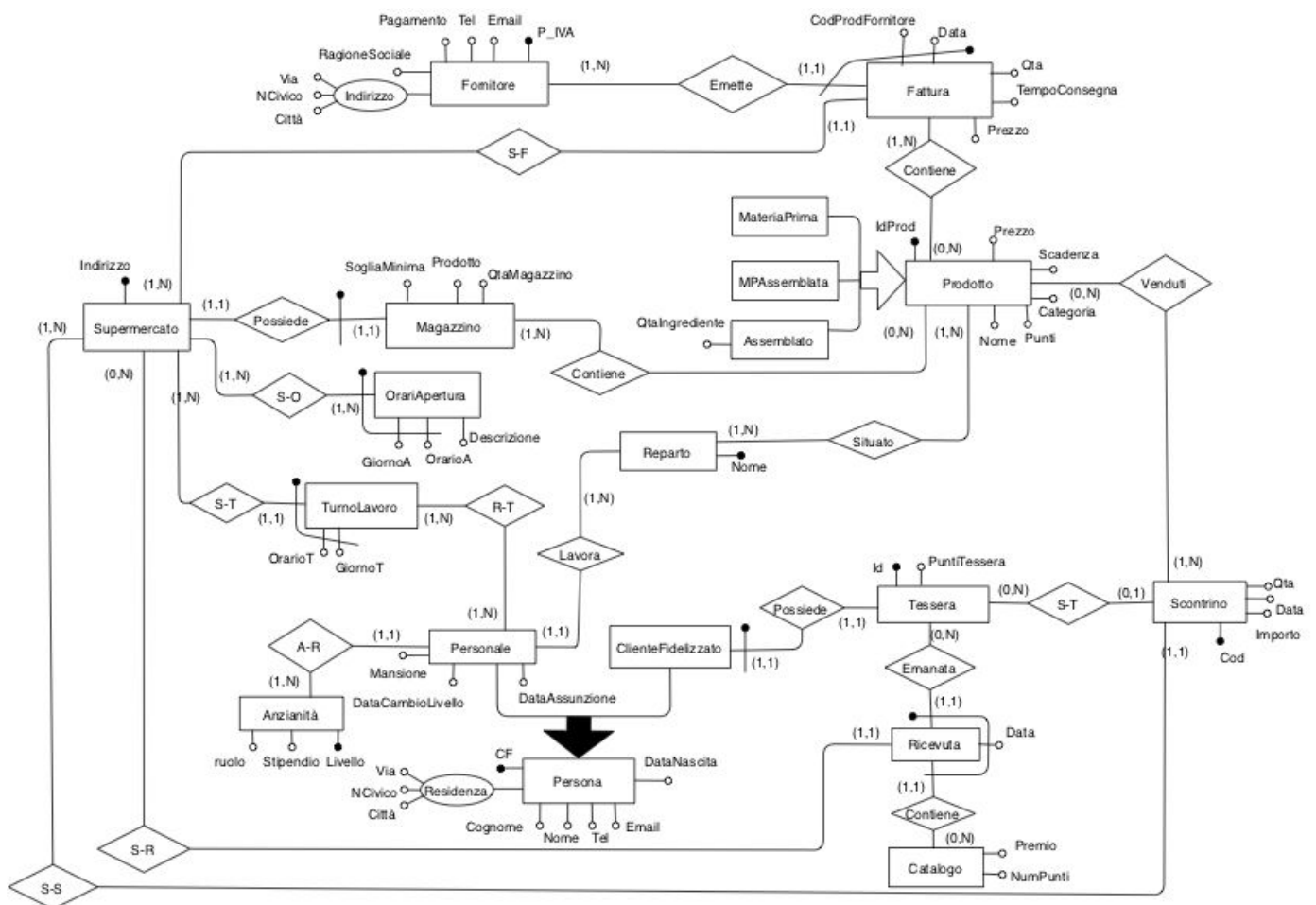
**Data di consegna: 20 luglio 2020**

**Studenti che intendono partecipare alla prova:**

**Giulia Scarpa**

**Il progetto è già stato consegnato in precedenza? No**

## 1.1 Schema ER



1. Supponiamo che nella città siano presenti 3 supermercati della catena ognuno con un proprio magazzino.
2. Gli orari di apertura possono essere ordinari o straordinari.
3. Ogni dipendente ha un particolare livello che ne determina l'anzianità di servizio, lo stipendio e il ruolo. Lo stipendio può variare dai 700 ai 1500€.
4. Assumiamo che i dipendenti possano lavorare per qualsiasi supermercato della catena.
5. Assumiamo di avere circa 20 fornitori.
6. Assumiamo di avere 16 reparti.
7. I prodotti potrebbero essere materie prime, assemblati, materie prime assemblate (che sono prodotti che il supermercato utilizza per produzione interna).
8. La quantità dello scontrino, del magazzino e della fattura devono sempre essere strettamente maggiori di zero. La stessa cosa vale per la soglia minima del magazzino.



(che a sua volta sono prodotti assemblati dal supermercato) oppure entrambe. La quantità dell'associazione ingrediente e dell'associazione composizione deve essere positiva, quindi maggiore uguale a 0.

8. La quantità dell'entità venduto, del magazzino e dell'associazione contiene devono sempre essere strettamente maggiori di zero. La stessa cosa vale per la soglia minima del magazzino.

9. Il prezzo del prodotto e il costo dell'associazione contiene devono essere sempre positivi quindi devono essere maggiori uguali a 0.

10. I punti tessera devono sempre essere maggiori uguali a 0 e minori uguali a 10000, mentre i punti del premio del catalogo devono essere strettamente maggiori di 0 e minori uguali a 10000. I punti del prodotto devono essere maggiori uguali a 0.

## 2.3 Modello relazionale

**Supermercato** (Indirizzo)

**OrariApertura** (IndirizzoA, GiornoA, OrarioA, Descrizione)

FOREIGN KEY (IndirizzoA) REFERENCES Supermercato (Indirizzo)

**Anzianità** (Livello, ruolo, Stipendio)

**Reparto** (Nome)

**Personale** (CE, Nome, Cognome, Tel, Email, Mansione, DataNascita, Residenza, DataAssunzione, DataCambioLivello, LivelloP, RepartoP)

FOREIGN KEY (LivelloP) REFERENCES Anzianità (Livello)

FOREIGN KEY (RepartoP) REFERENCES Reparto (Nome)

**TurnoLavoro** (cfT, GiornoT, OrarioT, IndirizzoT)

FOREIGN KEY (cfT) REFERENCES Personale (CF)

FOREIGN KEY (IndirizzoT) REFERENCES Supermercato (Indirizzo)

**ClienteFidelizzato** (IdTessera, CF, Nome, Cognome, TelC, Residenza, Email, DataNascita, NumPuntTot)

**Scontrino** (Cod, Data, IdTesseraS, IndirizzoS)

FOREIGN KEY (IdTesseraS) REFERENCES ClienteFidelizzato (IdTessera)

FOREIGN KEY (IndirizzoS) REFERENCES Supermercato (Indirizzo)

**Prodotto**(IdP, NomeP, PrezzoP, ScadenzaP, PuntiP, CategoriaP)

**MateriaPrima**(IdMP)

FOREIGN KEY (IdMP) REFERENCES Prodotto (IdP)

**Assemblato**(IdA)

FOREIGN KEY (IdA) REFERENCES Prodotto (IdP)

**MPAssemblata**(IdMPA)

FOREIGN KEY (IdMPA) REFERENCES Prodotto (IdP)

**Ingrediente**(IdAI, IdMPI, QtaMPI)

FOREIGN KEY (IdAI) REFERENCES Assemblato (IdA)

FOREIGN KEY (IdMPI) REFERENCES MateriaPrima (IdMP)

**Composizione**(IdAC, IdMPAC, QtaMPAC)

FOREIGN KEY (IdAC) REFERENCES Assemblato (IdA)

FOREIGN KEY (IdMPAC) REFERENCES MPAssemblata (IdMPA)

**Situato** (NomeRep, IdProdS)

FOREIGN KEY (NomeRep) REFERENCES Reparto (Nome)

FOREIGN KEY (IdProdS) REFERENCES Prodotto (IdP)

**Venduto**(IdPV, CodPS, QtaV)

FOREIGN KEY (IdPV) REFERENCES Prodotto (IdP)

FOREIGN KEY (CodPS) REFERENCES Scontrino (Cod)

**Magazzino** (IdProd, IndirizzoM, QtaTotale, SogliaMinima)

FOREIGN KEY (IdProd) REFERENCES Prodotto (IdP)

FOREIGN KEY (IndirizzoM) REFERENCES Supermercato (Indirizzo)

**Catalogo** (Premio, NumPunti)

**Ricevuta** (DataR, IdTesseraR, NomePremio, IndirizzoR)

FOREIGN KEY (IdTesseraR) REFERENCES ClienteFidelizzato (IdTessera)

FOREIGN KEY (NomePremio) REFERENCES Catalogo (Premio)

FOREIGN KEY (IndirizzoR) REFERENCES Supermercato (Indirizzo)

**Fornitore** (P\_IVA, Email, Tel, Indirizzo, RagioneSociale, Pagamento)

**Fattura** (Id, Data, TempoConsegna, P\_IVAF, IndirizzoF)

FOREIGN KEY (P\_IVAF) REFERENCES Fornitore (P\_IVA)

FOREIGN KEY (IndirizzoF) REFERENCES Fornitore (Indirizzo)

**Contiene** (IdProdC, IdFattura, CodProdFornitore, QtaC, Costo)

FOREIGN KEY (IdProdC) REFERENCES Prodotto (IdP)

FOREIGN KEY (IdFattura) REFERENCES Fattura (Id)

## 2.4 Codice SQL

```
--Supermercato (Indirizzo)
CREATE TABLE Supermercato (
    Indirizzo varchar(50) PRIMARY KEY
);

--OrariApertura (IndirizzoA, GiornoA, OrarioA, Descrizione)
CREATE TABLE OrariApertura (
    IndirizzoA varchar(50) NOT NULL REFERENCES Supermercato (Indirizzo)
ON DELETE CASCADE
ON UPDATE CASCADE,
    GiornoA varchar(50) NOT NULL,
    OrarioA varchar(50) NOT NULL,
    Descrizione varchar(50) NOT NULL,
    PRIMARY KEY(IndirizzoA, GiornoA, OrarioA)
);

--Anzianità(Livello, ruolo, Stipendio)
CREATE TABLE Anzianità (
    Livello char(1) PRIMARY KEY,
    ruolo varchar(50) NOT NULL,
    Stipendio numeric(4,0) NOT NULL,
    CONSTRAINT Anzianità_Stipendio_check CHECK ((Stipendio >= 700) AND (Stipendio <= 1500))
);

--Reparto (Nome)
CREATE TABLE Reparto (
    Nome varchar(50) PRIMARY KEY
```

);

--Personale (CE, Nome, Cognome, Tel, Email, Mansione, DataNascita, Residenza, DataAssunzione, DataCambioLivello, *LivelloP*, *RepartoP*)

```
CREATE TABLE Personale (  
    CF char(16) PRIMARY KEY,  
    Nome varchar(50) NOT NULL,  
    Cognome varchar(50) NOT NULL,  
    Tel varchar(10) NOT NULL UNIQUE,  
    Email varchar(50) NOT NULL UNIQUE,  
    Mansione varchar(80) NOT NULL,  
    DataNascita date NOT NULL,  
    Residenza varchar(50) NOT NULL,  
    DataAssunzione date NOT NULL,  
    DataCambioLivello date NOT NULL,  
    LivelloP char(1) NOT NULL REFERENCES Anzianità (Livello)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    RepartoP varchar(50) NOT NULL REFERENCES Reparto (Nome)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    UNIQUE(Nome, Cognome, DataNascita)  
);
```

--TurnoLavoro (cfT, GiornoT, OrarioT, IndirizzoT)

```
CREATE TABLE TurnoLavoro (  
    cfT char(16) NOT NULL REFERENCES Personale (CF)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    GiornoT varchar(50) NOT NULL,  
    OrarioT varchar(50) NOT NULL,  
    IndirizzoT varchar(50) NOT NULL REFERENCES Supermercato (Indirizzo)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    PRIMARY KEY(cfT, GiornoT, OrarioT, IndirizzoT)  
);
```

--ClienteFidelizzato (IdTessera, CF, Nome, Cognome, TelC, Residenza, Email, DataNascita, NumPuntTot)

```
CREATE TABLE ClienteFidelizzato (  
    IdTessera char(5) PRIMARY KEY,  
    CF char(16) UNIQUE,  
    Nome varchar(50) NOT NULL,  
    Cognome varchar(50) NOT NULL,  
    TelC varchar(10) NOT NULL,  
    Residenza varchar(50) NOT NULL,  
    Email varchar(50) NOT NULL,  
    DataNascita date NOT NULL,
```

```

        NumPuntiTot numeric(6,0) NOT NULL,
CONSTRAINT ClienteFidelizzato_NumPuntiTot_check CHECK ((NumPuntiTot>=0) AND
(NumPuntiTot<=10000))
);

```

```

--Scontrino (Cod, Data, IdTesseraS, IndirizzoS)
CREATE TABLE Scontrino (
    Cod varchar(10) PRIMARY KEY,
    Data date NOT NULL,
    IdTesseraS char(5) REFERENCES ClienteFidelizzato (IdTessera)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    IndirizzoS varchar(50) REFERENCES Supermercato (Indirizzo)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

--Prodotto(IdP, NomeP, PrezzoP, ScadenzaP, PuntiP, CategoriaP)
CREATE TABLE Prodotto (
    IdP char(5) UNIQUE NOT NULL PRIMARY KEY,
    NomeP varchar(50) UNIQUE NOT NULL,
    PrezzoP numeric(3,2) NOT NULL,
    ScadenzaP date NOT NULL,
    PuntiP numeric(3,0),
    CategoriaP varchar(50) NOT NULL,
CONSTRAINT Prodotto_PuntiP_check CHECK (PuntiP>=0),
CONSTRAINT Prodotto_PrezzoP_check CHECK (PrezzoP>0)
);

```

```

--MateriaPrima(IdMP)
CREATE TABLE MateriaPrima (
    IdMP char(5) NOT NULL REFERENCES Prodotto (IdP),
    PRIMARY KEY(IdMP)
);

```

```

--Assemblato(IdA)
CREATE TABLE Assemblato (
    IdA char(5) NOT NULL REFERENCES Prodotto (IdP),
    PRIMARY KEY(IdA)
);

```

```

--MPAssemblata(IdMPA)
CREATE TABLE MPAssemblata (
    IdMPA char(5) NOT NULL REFERENCES Prodotto (IdP),
    PRIMARY KEY(IdMPA)
);

```

```

--Ingrediente(IdAI, IdMPI, QtaMPI)

```

```

CREATE TABLE Ingrediente (
    IdAI char(5) NOT NULL REFERENCES Assemblato (IdA),
    IdMPI char(5) NOT NULL REFERENCES MateriaPrima (IdMP),
    QtaMPI numeric(2,0) NOT NULL,
    PRIMARY KEY(IdAI, IdMPI),
    CONSTRAINT Ingrediente_QtaMPI_check CHECK (QtaMPI >=0)
);

```

```

--Composizione(IdAC, IdMPAC, QtaMPAC)
CREATE TABLE Composizione (
    IdAC char(5) NOT NULL REFERENCES Assemblato (IdA),
    IdMPAC char(5) NOT NULL REFERENCES MPAssemblata (IdMPA),
    QtaMPAC numeric(2,0) NOT NULL,
    PRIMARY KEY(IdAC, IdMPAC),
    CONSTRAINT Composizione_QtaMPAC_check CHECK (QtaMPAC >=0)
);

```

```

--Situato (NomeRep, IdProdS)
CREATE TABLE Situato (
    NomeRep varchar(50) REFERENCES Reparto (Nome)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    IdProdS varchar(50) NOT NULL REFERENCES Prodotto (IdP)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    PRIMARY KEY(NomeRep, IdProdS)
);

```

```

--Venduto(IdPV, CodPS, QtaV)
CREATE TABLE Venduto (
    IdPV char(5) REFERENCES Prodotto (IdP)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CodPS varchar(10) NOT NULL REFERENCES Scontrino (Cod)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    QtaV numeric(2,0) NOT NULL,
    CONSTRAINT Venduto_QtaV_check CHECK (QtaV >0),
    PRIMARY KEY(IdPV, CodPS)
);

```

```

--Magazzino (IdProd, IndirizzoM, QtaTotale, SogliaMinima)
CREATE TABLE Magazzino (
    IdProdM char(5) NOT NULL REFERENCES Prodotto(IdP)
        ON DELETE CASCADE

```



```

        ON UPDATE CASCADE,
IndirizzoM varchar(50) REFERENCES Supermercato (Indirizzo)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
QtaTotale numeric(3,0),
SogliaMinima numeric(2,0),
PRIMARY KEY(IdProdM, IndirizzoM),
CONSTRAINT Magazzino_QtaTotale_check CHECK (QtaTotale > 0),
CONSTRAINT MateriaPrima_SogliaMinima_check CHECK (SogliaMinima>0)
);

--Catalogo(Premio, NumPunti)
CREATE TABLE Catalogo (
    Premio varchar(50) PRIMARY KEY,
    NumPunti numeric(6,0) NOT NULL,
    CONSTRAINT Catalogo_NumPunti_check CHECK CHECK ((NumPuntiTot>0) AND
(NumPuntiTot<=10000))
);

--Ricevuta (DataR, IdTesseraR, NomePremio, IndirizzoR)
CREATE TABLE Ricevuta (
    DataR date NOT NULL,
    IdTesseraR char(5) REFERENCES ClienteFidelizzato (IdTessera)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    NomePremio varchar(50) NOT NULL REFERENCES Catalogo (Premio)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    IndirizzoR varchar(50) REFERENCES Supermercato (Indirizzo)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    PRIMARY KEY(DataR, IdTesseraR, NomePremio)
);

--Fornitore(P_IVA, Email, Tel, Indirizzo, RagioneSociale, Pagamento)
CREATE TABLE Fornitore (
    P_IVA char(11) PRIMARY KEY,
    Email varchar(50) NOT NULL,
    Tel varchar(10) NOT NULL,
    Indirizzo varchar(50) NOT NULL,
    RagioneSociale varchar(50) NOT NULL,
    Pagamento varchar(50) NOT NULL,
    UNIQUE(Email, Tel, Indirizzo)
);

--Fattura (Id, Data, TempoConsegna, P_IVAF, IndirizzoF)
CREATE TABLE Fattura (

```

```

    Id char(10) PRIMARY KEY,
    Data date NOT NULL,
    TempoConsegna varchar(2) NOT NULL,
    P_IVAF char(11) REFERENCES Fornitore (P_IVA)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    IndirizzoF varchar(50) REFERENCES Supermercato (Indirizzo)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

--Contiene (IdProdC, IdFattura, CodProdFornitore, QtaC, Costo)
CREATE TABLE Contiene (
    IdProdC varchar(5) NOT NULL REFERENCES Prodotto (IdP)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    IdFattura varchar(10) NOT NULL REFERENCES Fattura (Id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CodProdFornitore char(6) NOT NULL,
    QtaC numeric(3,0) NOT NULL,
    Costo numeric(4,2) NOT NULL,
    PRIMARY KEY(IdProdC, IdFattura),
    CONSTRAINT Contiene_QtaC_check CHECK (QtaC > 0),
    CONSTRAINT Contiene_Costo_check CHECK (Costo >= 0)
);

```

## OPERAZIONI:

### --Inserimento e modifica di un reparto

```

INSERT INTO Reparto VALUES ('Casse');
UPDATE Reparto SET nome='Cassa' WHERE nome='Casse';

```

### --Inserimento e modifica del responsabile e degli impiegati di un reparto

```

INSERT INTO Personale (cf, mansione, repartop) VALUES ('PLRFNN69B22F205Z', 'Capo Reparto', 'Frutta e
verdura');
INSERT INTO Personale (cf, mansione, repartop) VALUES ('FRRGLR91D61F205T', 'Verifica delle scadenze',
'Salumi e formaggi');
UPDATE Personale SET repartop = 'Casse' WHERE cf = 'FRRGLR91D61F205T' ;
UPDATE Personale SET mansione = 'Attività di inventario' WHERE cf = 'FRRGLR91D61F205T' ;

```

### --Visualizzazione del personale in organico in un reparto

```

SELECT *
FROM Personale
WHERE repartop='Casse'

```

**--Visualizzazione delle mansioni in uso nel reparto e, per ogni mansione, del personale del reparto che copre quella mansione**

```
SELECT cf, mansione  
FROM personale  
WHERE repartop = 'Frutta e verdura'
```

```
SELECT cf, mansione  
FROM personale  
WHERE repartop = 'Salumi e formaggi' AND mansione='Verifica delle scadenze'
```

**--Visualizzazione dei turni settimanali del personale del reparto**

```
SELECT cft, giornot, orariot, indirizzot  
FROM turnolavoro JOIN personale ON cft=cf  
WHERE repartop='Frutta e verdura'
```

**--Interrogazioni: determinare le vendite nel reparto macelleria effettuate in un supermercato nella giornata del 12 maggio del 2020**

```
SELECT idpv, SUM(qtav)  
FROM situato INNER JOIN venduto ON idprods=idpv  
INNER JOIN scontrino ON codps=cod  
WHERE nomerep='Macelleria' AND data='2020-05-12'  
GROUP BY idpv
```

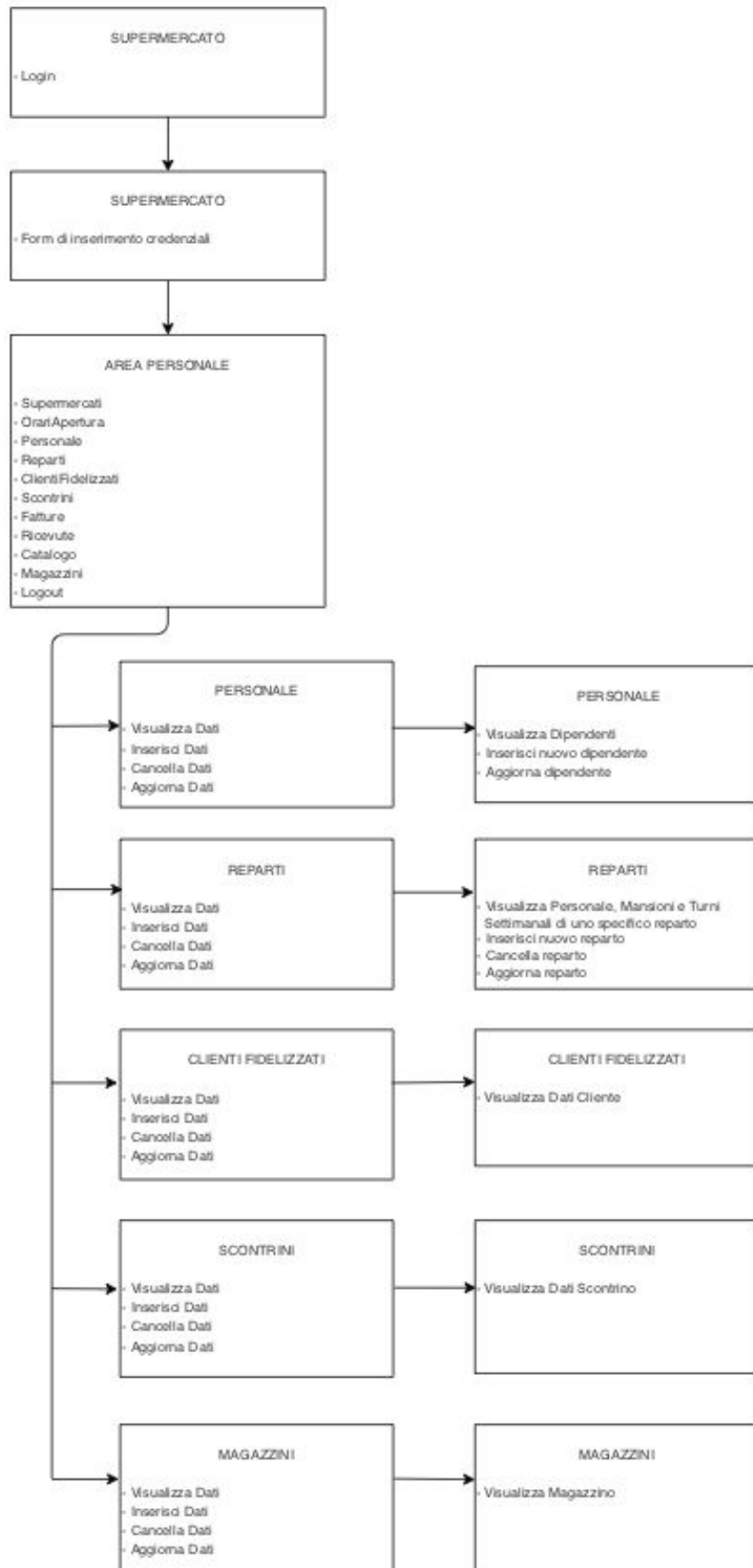
**--Interrogazioni: determinare per i prodotti del reparto cancelleria che richiedono di essere riordinati il tempo minimo di riapprovvigionamento**

```
SELECT tempoconsegna  
FROM situato  
INNER JOIN contiene ON idprods=idprodc  
INNER JOIN fattura ON idfattura=id  
WHERE nomerep='Cancelleria'
```

**--Interrogazioni: determinare i prodotti preparati nel supermercato assemblati usando almeno un altro prodotto preparato nel supermercato**

```
SELECT DISTINCT idac  
FROM composizione
```

### 3. Progettazione del sito



Nome del database PostgreSQL: supermercato