

La Rinascente

Progetti di Basi di Dati

A.A. 2017/2018

Giovanni Peron

Matr:1137766

Samuele Gardin

Matr:1143807

Indice

Abstract	4
Analisi dei requisiti	5
Progettazione concettuale	7
ENTITÀ e relativi ATTRIBUTI:	7
RELAZIONI e RELATIVI ATTRIBUTI:.....	8
GENERALIZZAZIONI	9
REGOLE DI VINCOLO	9
Progettazione logica	10
SCHEMA ER RISTRUTTURATO	10
ENTITÀ e relativi ATTRIBUTI:	11
RELAZIONI e RELATIVI ATTRIBUTI:.....	12
SCHEMA RELAZIONALE	14
Implementazione.....	15
QUERY E PROCEDURE	15
TRIGGER.....	20
FUNZIONI	24

Abstract

In questo progetto verrà modellata una base di dati per una casa di riposo. Il database servirà per gestire le interazioni tra il personale della struttura e il magazzino, in particolare quelle con la cucina. Infatti, uno degli aspetti principali del progetto sarà quello di fornire un supporto per memorizzare un ricettario. Il ricettario sarà uno strumento utile per il personale della cucina, perché permetterà ad ognuno di visualizzare per ogni ricetta tutti i suoi ingredienti con la relativa quantità. È previsto anche l'inserimento di una procedura che permetta ai cuochi di calcolare la quantità di ogni ingrediente di una certa ricetta in base al numero di persone per le quali si deve preparare tale ricetta. Inoltre, il database servirà a gestire tutte le richieste di articoli che il personale della struttura rivolge ai magazzinieri e anche tutte le ordinazioni che i magazzinieri propongono quando si trovano nella necessità di rifornire il magazzino con un determinato articolo. Le altre caratteristiche significative della base di dati in questione sono:

- il mantenimento di uno schedario contenente le informazioni per tutti gli articoli presenti in magazzino;
- la memorizzazione di un elenco dei fornitori che ricevono ordinazioni dalla struttura, con relative informazioni per ognuno di essi.

Analisi dei requisiti

Si vuole creare una base di dati per la casa di riposo: “*La Rinascente*”, che desidera introdurre un ricettario consultabile dal personale della cucina e dai magazzinieri. Contemporaneamente è richiesta la gestione dell'organizzazione tra il magazzino e il personale della struttura, questa consiste in delle richieste, relative a determinati articoli, rivolte ai magazzinieri da parte di tutto il resto del personale.

Il personale di interesse sarà composto da cuochi, magazzinieri e il resto dei dipendenti, di ognuno conosciamo:

- un codice di identificazione;
- nome;
- cognome;
- codice fiscale;
- numero di telefono;
- email (opzionale).

I magazzinieri ricevono richieste da tutto il personale (esclusi altri magazzinieri), per ogni richiesta conosceremo:

- il nome dell'articolo;
- la quantità necessaria;
- un'unità di misura;
- la data della richiesta.

Il magazzino è diviso per categoria di articoli, ognuna gestita da un magazziniere. Ogni categoria potrebbe essere composta da una o più stanze in diversi piani. Per cui i dati che identificano un magazzino sono:

- numero di stanza;
- piano.

Quando un articolo è in carenza il magazziniere fa un ordine per ogni ordine è necessario conoscere:

- il prodotto ordinato;
- la quantità del prodotto ordinato;
- un'unità di misura;
- la data dell'ordine.

Il magazziniere non fa un vero e proprio ordine ma una proposta che verrà presentata e poi concretizzata dall'ufficio economato.

Gli ordini sono riferiti ad un fornitore del quale vogliamo sapere:

- piva;
- denominazione;
- numero di telefono;
- email (opzionale).

Ogni magazzino contiene diversi articoli divisi appunto in categorie: alimenti, detersivi, teleria, generici. Tutti gli articoli hanno le seguenti informazioni:

- un codice identificativo;
- un nome;
- la partita o il lotto di produzione (opzionale);
- la quantità presente in magazzino;
- un'unità di misura.

Esistono alimenti deperibili e a lunga conservazione, per entrambi bisogna conoscere:

- data di scadenza (opzionale);
- il termine minimo di conservazione (opzionale);
- data di produzione o di confezionamento (opzionale).

Tutte queste informazioni relative agli alimenti sono tutte opzionali perché tutti i prodotti alimentari dovrebbero possederle ma non è detto che le abbiano tutte.

Gli alimenti deperibili devono essere conservati in celle frigorifere, ognuna identificata da un numero. Ci sono quattro tipi di celle: verdura, salumi, formaggi, surgelati. Quindi sapremo in che cella trovare ogni alimento deperibile.

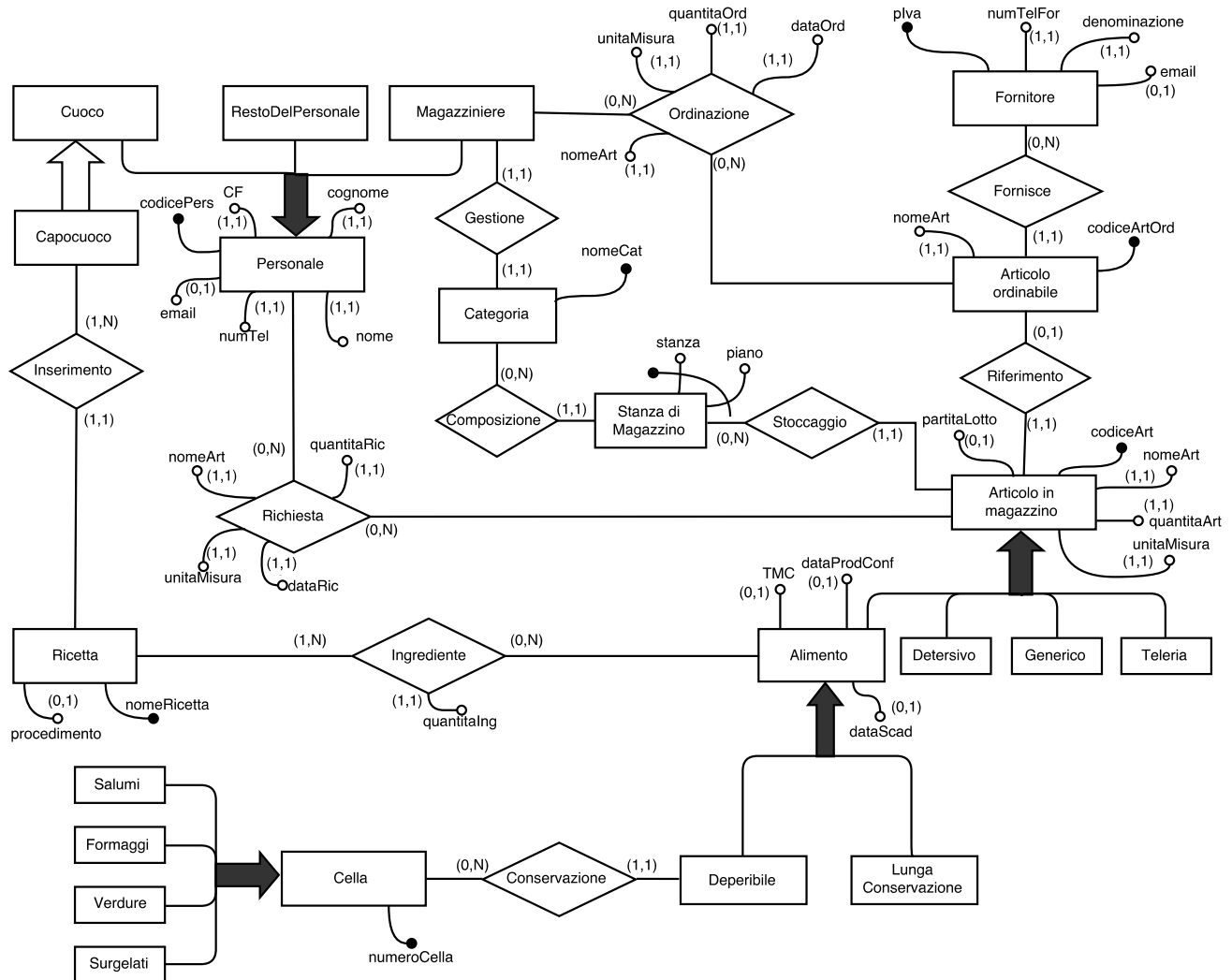
Infine, i cuochi e i magazzinieri possono consultare un ricettario nel quale solo il capocuoco dovrà inserire nuove ricette. Il ricettario conterrà più ricette e per ogni ricetta sarà noto:

- un nome;
- un procedimento (opzionale).

Ogni ricetta avrà un certo numero di ingredienti, presenti tra gli alimenti del magazzino, con relativa quantità che potrebbe essere espressa in diverse unità di misura.

Progettazione concettuale

SCHEMA ER



ENTITÀ e relativi ATTRIBUTI:

- Personale
 - *codicePers* : int (identificatore)
 - *nome* : varchar(15)
 - *cognome* : varchar(15)
 - *CF*: char(16)
 - *numTel* : varchar(14)
 - *email* (opzionale) : varchar(50)
- Categoria
 - *nomeCat* : varchar(25) (identificatore)
- Magazzino
 - *stanza* : int (identificatore)
 - *piano* : int (identificatore)

- Fornitore
 - plva : char(11) (identificatore)
 - denominazione : varchar(50)
 - numTelFor : varchar(14)
 - email (opzionale) : varchar(50)
- Articolo ordinabile
 - codiceArtOrd : varchar(15) (identificatore)
 - nomeArt : varchar(25)
- Articolo in magazzino
 - codiceArt : varchar(15) (identificatore)
 - nomeArt : varchar(25)
 - quantitaArt : int
 - partitaLotto (opzionale) : int
 - unitaMisura : enum('KG','NR','LT','GR')
- Ricetta
 - nomeRicetta : varchar(25) (identificatore)
 - procedimento : mediumtext
- Cella
 - numeroCella : int (identificatore)

RELAZIONI e RELATIVI ATTRIBUTI:

- Personale - Articolo in magazzino: **Richiesta**
 - Ogni membro del personale può richiedere uno o più articoli del magazzino
 - Ogni articolo del magazzino può essere richiesto da uno o più membri del personale
 - nomeArt : varchar(25)
 - quantitaRic : int
 - dataRic : date
 - unitaMisura : enum('KG','NR','LT','GR')
- Magazziniere - Articolo ordinabile: **Ordinazione**
 - Ogni magazziniere può ordinare uno o più articoli ordinabili
 - Ogni articolo ordinabile può essere ordinato da uno o più magazzinieri (effettivamente il magazziniere sarà sempre lo stesso perché esiste un solo magazziniere per ogni categoria)
 - nomeArt : varchar(25)
 - quantitaOrd : int
 - dataOrd : date
 - unitaMisura : enum('KG','NR','LT','GR')
- Fornitore - Articolo ordinabile: **Fornisce**
 - Ogni fornitore può fornire uno o più articoli ordinabili
 - Ogni articolo ordinabile viene fornito da un solo fornitore
- Magazziniere - Categoria: **Gestione**
 - Ogni magazziniere gestisce una sola categoria del magazzino
 - Ogni categoria del magazzino viene gestita da un solo magazziniere
- Categoria - Stanza di Magazzino: **Composizione**
 - Ogni categoria può essere composta da una o più stanze del magazzino
 - Ogni stanza del magazzino appartiene ad una sola categoria

- Stanza di Magazzino - Articolo in magazzino: **Stoccaggio**
 - In ogni stanza di magazzino vengono stoccati uno o più articoli, la stanza potrebbe però essere momentaneamente vuota
 - Ogni articolo è stoccato in una stanza del magazzino
- Capocuoco - Ricetta: **Inserimento**
 - Il capocuoco inserisce una o più ricette
 - Una ricetta viene inserita solo dal capocuoco
- Ricetta - Alimento: **Ingrediente**
 - Ogni ricetta ha come ingrediente uno o più alimenti
 - Ogni alimento può essere ingrediente di una o più ricette
 - quantalng : int
- Cella - Deperibile: **Conservazione**
 - Ogni alimento deperibile è conservato in una cella frigorifera, la cella potrebbe però essere momentaneamente vuota
 - Ogni cella frigorifera conserva uno o più alimenti deperibili

GENERALIZZAZIONI

- Ogni membro dell'entità personale o è un cuoco o un magazziniere oppure fa parte del resto del personale
 - Solo un membro dell'entità cuoco è il capocuoco
- Ogni elemento dell'entità articolo in magazzino è un alimento o un detersivo o un articolo generico o un articolo di teleria.
 - A sua volta ogni alimento è deperibile o a lunga conservazione. Ed ha come attributi:
 - TMC : date, (termine minimo di conservazione)
 - dataProdConf : date, (data di produzione o di confezionamento)
 - dataScad : date, (data di scadenza)
- Ogni cella frigorifera è una cella per salumi o per formaggi o per verdure oppure per alimenti surgelati.

REGOLE DI VINCOLO

RV1 - Un magazziniere non deve ricevere richieste da un altro magazziniere, più in generale un magazziniere non deve fare richieste.

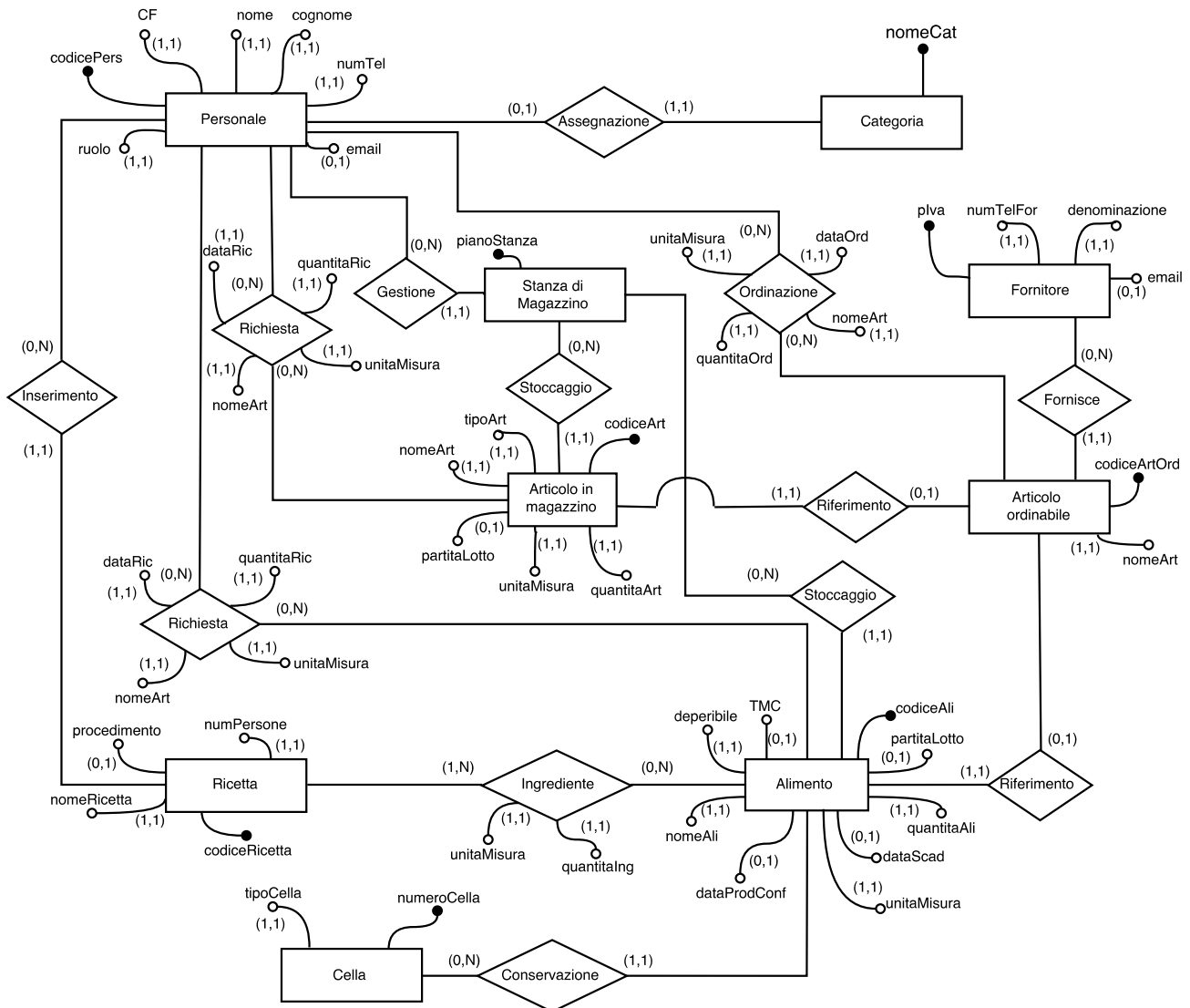
RV2 - Una stanza di magazzino deve contenere tutti articoli appartenenti alla stessa categoria.

RV3 - La quantità di un articolo specificata in una richiesta deve essere minore o uguale alla quantità disponibile in magazzino per quell'articolo.

RV4 - Una ricetta deve essere inserita da un membro del personale con il ruolo corrispondente a capocuoco.

Progettazione logica

SCHEMA ER RISTRUTTURATO



Nella fase di ristrutturazione abbiamo risolto le varie generalizzazioni:

- **Personale**: Ogni membro dell'entità personale o è un cuoco o un magazziniere oppure fa parte del resto del personale, solo un membro dell'entità cuoco è il capocuoco. Questa generalizzazione è stata ristrutturata includendo nell'entità Personale le entità RestoDelPersonale, Cuoco (e Capocuoco, che era generalizzazione di Cuoco), e Magazziniere. Per fare ciò abbiamo aggiunto a Personale l'attributo ruolo che potrà contenere un valore che identificherà il ruolo di ogni membro del personale.
- **Articolo in magazzino**: Ogni elemento dell'entità articolo in magazzino è un alimento o un detersivo o un articolo generico o un articolo di teleria. A sua volta ogni alimento è deperibile o a lunga conservazione. Per questa generalizzazione abbiamo incluso le entità Generico, Teleria, e Detersivo in Articolo aggiungendo un attributo tipoArt per identificare queste categorie. Abbiamo invece lasciato un'entità Alimento perché ipotizziamo che gli accessi ad alimento saranno separati dagli accessi all'entità Articolo. Inoltre nell'entità Alimento abbiamo

aggiunto l'attributo deperibile : bool per identificare se un alimento è deperibile o a lunga conservazione.

- Cella: Ogni cella frigorifera è una cella per salumi o per formaggi o per verdure oppure per alimenti surgelati. Questa generalizzazione è stata risolta includendo in Cella l'attributo tipoCella che identifica se la cella è per salumi, formaggi, verdure o surgelati.

Inoltre, abbiamo associato l'entità Categoria solamente con l'entità Magazziniere (accorpata ora a Personale), ottenendo la relazione Personale - Categoria: **Assegnazione**. Ogni membro del personale con ruolo magazziniere è assegnato ad un'unica categoria, e ogni categoria è assegnata ad un magazziniere. Mentre la relazione Categoria - Stanza di Magazzino: **Composizione** è stata sostituita dalla relazione Personale - Stanza di magazzino: **Gestione**, in quanto avendo magazziniere assegnato ad una categoria e sapendo quale stanza di magazzino esso gestisce, possiamo dedurre di che categoria è la stanza di magazzino.

ENTITÀ e relativi ATTRIBUTI:

- Personale
 - codicePers : int (identificatore)
 - nome : varchar(15)
 - cognome : varchar(15)
 - CF: char(16)
 - numTel : varchar(14)
 - email (opzionale) : varchar(50)
 - ruolo:enum('magazziniere','cuoco','capoCuoco','operatore','servizioGenerale','direttore','altro')
- StanzeDiMagazzino
 - pianoStanza : varchar(5) (identificatore)
- Categoria
 - NomeCat : varchar(15) (identificatore)
- Fornitore
 - plva : char(11) (identificatore)
 - denominazione : varchar(50)
 - numTelFor : varchar(14)
 - email : varchar(50)
- Articolo in magazzino
 - codiceArt : varchar(15) (identificatore)
 - nomeArt : varchar(25)
 - quantitaArt : int
 - unitaMisura : enum('KG', 'NR', 'LT', 'GR')
 - tipoArt : enum('alimento','generico','detersivo','teleria')
 - partitaLotto : int
- Articolo ordinabile
 - codiceArtOrd : varchar(15) (identificatore)
 - nomeArt : varchar(25)
- Cella
 - numeroCella : int (identificatore)
 - tipoCella : enum('salumi','verdure','formaggi','surgelati')

- Alimento
 - codiceAli: varchar(15) (identificatore)
 - nomeAli : varchar(25)
 - quantitaAli : int
 - unitaMisura : enum('KG', 'NR', 'LT', 'GR')
 - deperibile: bool
 - dataScad: date
 - TMC : date, (termine minimo di conservazione)
 - dataProdConf : date, (data di produzione o di confezionamento)
 - partitaLotto : int
- Ricetta
 - codiceRicetta : varchar(15) (identificatore)
 - nomeRicetta : varchar(25)
 - procedimento : mediumtext
 - numPersone : int

RELAZIONI e RELATIVI ATTRIBUTI:

- Ricetta - Alimento: **Ingrediente**
 - Ogni ricetta ha come ingrediente uno o più alimenti
 - Ogni alimento può essere ingrediente di una o più ricette
 - quantitaIng : int
 - unitaMisura : enum('KG', 'NR', 'LT', 'GR')
- Personale - Articolo ordinabile: **Ordinazione**
 - Ogni membro del personale con ruolo magazziniere può ordinare uno o più articoli ordinabili
 - Ogni articolo ordinabile può essere ordinato da uno o più magazzinieri (effettivamente il magazziniere sarà sempre lo stesso perché esiste un solo magazziniere per ogni categoria)
 - quantitaOrd : int
 - unitaMisura : enum('KG', 'NR', 'LT', 'GR')
 - dataOrd : date
 - nomeArt : varchar(25)
- Personale - Articolo in magazzino: **Richiesta**
 - Ogni membro del personale può richiedere uno o più articoli del magazzino
 - Ogni articolo del magazzino può essere richiesto da uno o più membri del personale
 - nomeArt : varchar(25)
 - dataRic : date
 - quantitaRic : int
 - unitaMisura : enum('KG', 'NR', 'LT', 'GR')
- Personale - Alimento: **Richiesta**
 - Ogni membro del personale può richiedere uno o più alimenti
 - Ogni alimento può essere richiesto da uno o più membri del personale
 - nomeArt : varchar(25)
 - dataRic : date
 - quantitaRic : int
 - unitaMisura : enum('KG', 'NR', 'LT', 'GR')

- Personale - Ricetta: **Inserimento**
 - Il membro del personale con ruolo capocuoco inserisce una o più ricette
 - Una ricetta viene inserita solo dal capocuoco
- Personale - Stanza di magazzino: **Gestione**
 - Ogni membro del personale con ruolo magazziniere gestisce più stanze di una sola categoria del magazzino
 - Ogni stanza di magazzino viene gestita da un solo magazziniere, e quella stanza sarà per forza della categoria assegnata al magazziniere
- Fornitore - Articolo ordinabile: **Fornisce**
 - Ogni fornitore può fornire uno o più articoli ordinabili
 - Ogni articolo ordinabile viene fornito da un solo fornitore
- Stanza di magazzino - Articolo in magazzino: **Stoccaggio**
 - In ogni stanza di magazzino vengono stoccati uno o più articoli, la stanza potrebbe però essere momentaneamente vuota
 - Ogni articolo è stoccato in una stanza del magazzino
- Articolo ordinabile - Alimento: **Riferimento**
 - Ogni articolo ordinabile può riferirsi ad un alimento
 - Ogni alimento è riferito ad un articolo ordinabile
- Articolo ordinabile - Articolo in magazzino: **Riferimento**
 - Ogni articolo ordinabile può riferirsi ad un articolo in magazzino
 - Ogni articolo in magazzino è riferito ad un articolo ordinabile
- Alimento - Stanza di magazzino: **Stoccaggio**
 - Ogni alimento è stoccato in una stanza del magazzino
 - In ogni stanza del magazzino vengono stoccati uno o più alimenti, la stanza potrebbe però essere momentaneamente vuota
- Alimento - Cella: **Conservazione**
 - Ogni alimento è conservato in una cella frigorifera
 - Ogni alimento deperibile è conservato in una cella frigorifera, la cella potrebbe però essere momentaneamente vuota
- Personale - Categoria: **Assegnazione**
 - Ogni membro del personale con ruolo magazziniere è assegnato ad un'unica categoria
 - Ogni categoria è assegnata ad un magazziniere

SCHEMA RELAZIONALE

Nel seguente schema relazionale le chiavi primarie sono in grassetto e sottolineate, mentre le chiavi esterne sono in corsivo e sotto alla corrispondente relazione viene specificato a quale attributo si riferiscono.

Cella (**numeroCella**, tipoCella)

Personale (**codicePers**, nome, cognome, CF, numTel, email, ruolo)

StanzaDiMagazzino (**pianoStanza**, *codiceMgznr*)

codiceMgznr FK(**Personale**(*codicePers*))

Fornitore (**plva**, denominazione, numTelFor, email)

ArticoloOrdinabile (**codiceArtOrd**, nomeArt, *plvaFornitore*)

plvaFornitore FK(**Fornitore** (*plva*))

Alimento (**codiceAli**, nomeAli, quantitaAli, unitaMisura, deperibile, dataScad, TMC, dataProdConf, partitaLotto, *pianoStanza*, *numeroCella*, *codiceArtOrd*)

pianoStanza FK(**StanzaDiMagazzino**(*pianoStanza*))

numeroCella FK(**Cella**(*numeroCella*))

codiceArtOrd FK(**ArticoloOrdinabile**(*codiceArtOrd*))

ArticoloMag (**codiceArtMag**, nomeArt, quantitaArt, unitaMisura, tipoArt, partitaLotto, *pianoStanza*, *codiceArtOrd*)

pianoStanza FK(**StanzaDiMagazzino**(*pianoStanza*))

codiceArtOrd FK(**ArticoloOrdinabile**(*codiceArtOrd*))

Categoria (**nomeCat**, *codiceMgznr*)

codiceMgznr FK(**Personale**(*codicePers*))

Ricetta (**codiceRicetta**, nomeRicetta, procedimento, numPersone, *codiceCapoCuoco*)

codiceCapoCuoco FK(**Personale**(*codicePers*))

Ingrediente (**codiceIng**, *codiceRicetta*, *codiceAli*, quantitaIng, unitaMisura)

codiceRicetta FK(**Ricetta** (*codiceRicetta*))

codiceAli FK(**Alimento**(*codiceAli*))

Ordinazione (**codiceOrd**, *codiceMgznr*, *codiceArtOrd*, nomeArt, dataOrd, quantitaOrd, unitaMisura)

codiceMgznr FK(**Personale**(*codicePers*))

codiceArtOrd FK(**ArticoloOrdinabile**(*codiceArtOrd*))

RichiestaAlimento (**codiceRic**, *codicePers*, *codiceAli*, nomeArt, dataRic, quantitaRic, unitaMisura)

codicePers FK(**Personale**(*codicePers*))

codiceAli FK(**Alimento**(*codiceAli*))

RichiestaArtMag (**codiceRic**, *codicePers*, *codiceArtMag*, nomeArt, dataRic, quantitaRic, unitaMisura)

codicePers FK(**Personale**(*codicePers*))

codiceArtMag FK(**ArticoloMag**(*codiceArtMag*))

Implementazione

QUERY E PROCEDURE

1. Query che ritorni il codice e il nome di tutti gli articoli in magazzino di tipo detersivo con le denominazioni dei relativi fornitori e con la quantità ordinata che dev'essere maggiore di 5 NR.

```
SELECT ao.codiceArtOrd, am.codiceArtMag, am.nomeArt, o.quantitaOrd,
f.denominazione, f.pIva
FROM articoloordinabile ao
JOIN ordinazione o ON (o.codiceArtOrd = ao.codiceArtOrd)
JOIN fornitore f ON (f.pIva = ao.pIvaFornitore)
JOIN articoloMag am ON (am.codiceArtOrd = ao.codiceArtOrd)
WHERE o.quantitaOrd > 5 AND o.unitaMisura = 'NR'
AND am.tipoArt = "detersivi";
```

Output della query:

codiceArtOrd	codiceArtMag	nomeArt	quantitaOrd	denominazione	pIva
7	D0001	SAPONE MANI DERMOMED NR=5LT	6	ITALCHIMICA SRL	26971442321
8	D0002	SANITEC LAVAPIATTI NR=5LT	10	ITALCHIMICA SRL	26971442321
11	D0003	LAVAVETRI	8	Alchimia Detergenti	38544768294

2. Query che ritorni l'e-mail dei fornitori che non hanno mai partecipato ad ordinazioni eseguite dal magazziniere della categoria alimentari

```
SELECT email
FROM fornitore
WHERE pIva NOT IN (
    SELECT ao.pIvaFornitore
    FROM ordinazione o
    JOIN articoloordinabile ao ON o.codiceArtOrd = ao.codiceArtOrd
    WHERE o.codiceMgznr = (
        SELECT codicePers
        FROM personale p
        JOIN categoria c ON p.codicePers = c.CodiceMgznr
        WHERE p.ruolo = "magazziniere" AND c.nomeCat = "alimenti"
    )
);
```

Output della query:

email
daFrom.srl@gmail.com
carraro.caffe@alice.it
dolciariaFGF@gmail.com
ortofrutta.squarcina@hotmail.it
info@italchimica.it
alchimia@hotmail.it
tessility@yahoo.com
hds-foodserviceROBO@gmail.com
longhinFabiano@libero.it

3. Query che ritorni nome e cognome dei membri del personale che hanno fatto una richiesta al magazziniere Giovanni Storti nell'autunno di quest'anno, specificando il nome dell'articolo richiesto

```
CREATE VIEW Query3_richieste AS
SELECT nomeArt, codicePers, codiceAli AS codiceArt, dataRic
FROM richiestaalimento
UNION ALL
SELECT nomeArt, codicePers, codiceArtMag AS codiceArt, dataRic
FROM richiestaartmag;
```

```
CREATE VIEW Query3_articoli AS
SELECT codiceAli AS codiceArt, pianoStanza
FROM `alimento`
UNION ALL
SELECT codiceArtMag AS codiceArt, pianoStanza
FROM articolomag;
```

```
CREATE VIEW Query3 AS
SELECT p.nome, p.cognome, r.nomeArt
FROM personale p
JOIN Query3_richieste r ON (p.codicePers = r.codicePers)
JOIN Query3_articoli art ON (r.codiceArt = art.codiceArt)
JOIN stanzadimagazzino s ON (s.pianoStanza = art.pianoStanza)
WHERE r.dataRic
BETWEEN concat(YEAR(CURDATE()), '-09-23')
AND concat(YEAR(CURDATE()), '-12-21')
AND s.codiceMgznr = (
    SELECT codicePers
    FROM personale
    WHERE nome = 'giovanni' AND cognome = 'storti'
) ORDER BY p.nome, p.cognome;
```


Output della query:

nome	cognome	nomeArt
Carlo	Bianchi	LATTE UHT P.S. LT 1 NR=1LT
Claudio	Penna	BIETA ERBETTA GELO KG 2,5
Giovanna	Minto	VINO BIANCO LT 1 NR=LT 1
Paola	Rossi	CAROTA NR=CASSETTA DA 160

4. Query che restituisca il nome delle ricette che hanno ingredienti riferiti ad alimenti che sono scaduti nella cella dei formaggi

```
SELECT r.nomeRicetta, a.nomeAli AS Scaduto
FROM ricetta r
JOIN ingrediente i ON (r.codiceRicetta = i.codiceRicetta)
JOIN alimento a ON (i.codiceAli = a.codiceAli)
WHERE dataScad < CURDATE() AND numeroCella = (
    SELECT numeroCella
    FROM cella
    WHERE tipoCella = 'formaggi'
);
```

Output della query:

nomeRicetta	Scaduto
SPAGHETTI ALLA CARBONARA	LATTE UHT P.S. LT 1 NR=1LT
BACCALA' ALLA VICENTINA	LATTE UHT P.S. LT 1 NR=1LT
CAROTE TRIFOLATE	BURRO KG 1
PENNE AL PESTO GENOVESE	BURRO KG 1
RISOTTO AL RADICCHIO DI TREVISO	BURRO KG 1

5. Query che ritorni la media del numero di richieste effettuate negli ultimi 30 giorni

```
CREATE VIEW Query5_richieste AS
SELECT nomeArt, codicePers, codiceAli AS codiceArt, dataRic
FROM richiestaalimento
UNION ALL
SELECT nomeArt, codicePers, codiceArtMag AS codiceArt, dataRic
FROM richiestartaartmag;

CREATE VIEW Query5_numRicUlt30 AS
SELECT COUNT(*) AS num
FROM Query5_richieste AS r
WHERE r.dataRic
BETWEEN (CURDATE() - INTERVAL 1 MONTH ) AND CURDATE();

CREATE VIEW Query5 AS
SELECT num / 30 AS richiesteMedieUltimi30gg
FROM Query5_numRicUlt30;
```

Output della query:

richiesteMedieUltimi30gg
0.0667

6. Query che restituisca il nome del magazziniere che gestisce più articoli tenendo conto anche degli alimenti

```
CREATE VIEW Query6_conteggio AS
    SELECT COUNT(*) numeroArt, c.nomeCat, p.nome, p.cognome
    FROM personale p, stanzadimagazzino s, articolomag a, categoria c
    WHERE s.codiceMgznr = p.codicePers AND c.nomeCat = 'detersivi'
    AND s.pianoStanza = a.pianoStanza AND p.codicePers = c.codiceMgznr
    UNION ALL
    SELECT COUNT(*) numeroArt, c.nomeCat, p.nome, p.cognome
    FROM personale p, stanzadimagazzino s, articolomag a, categoria c
    WHERE s.codiceMgznr = p.codicePers AND c.nomeCat = 'teleria'
    AND s.pianoStanza = a.pianoStanza AND p.codicePers = c.codiceMgznr
    UNION ALL
    SELECT COUNT(*) numeroArt, c.nomeCat, p.nome, p.cognome
    FROM personale p, stanzadimagazzino s, articolomag a, categoria c
    WHERE s.codiceMgznr = p.codicePers AND c.nomeCat = 'generici'
    AND s.pianoStanza = a.pianoStanza AND p.codicePers = c.codiceMgznr
    UNION ALL
    SELECT COUNT(*) numeroArt, c.nomeCat, p.nome, p.cognome
    FROM personale p, stanzadimagazzino s, alimento a, categoria c
    WHERE s.codiceMgznr = p.codicePers AND c.nomeCat = 'alimenti'
    AND s.pianoStanza = a.pianoStanza AND p.codicePers = c.codiceMgznr;

SELECT nome, cognome
FROM Query6_conteggio
WHERE numeroArt = (
    SELECT MAX(numeroArt)
    FROM Query6_conteggio
) ;
```

Output della query:

nome	cognome
giovanni	storti

PROCEDURE

1. Procedura che data una ricetta e un numero di persone ritorni la lista degli ingredienti con la loro quantità modificata in modo proporzionale al numero di persone specificato alla funzione

```
CREATE PROCEDURE ricettaPerNPersone(IN nPersone INT, IN codRicetta
VARCHAR(15))
BEGIN
    DECLARE nP INT;
    SET nP = (
        SELECT numPersone
        FROM ricetta
        WHERE codiceRicetta = codRicetta
    );
```

```

SELECT a.nomeALi,
cast((i.quantitaIng*nPersone) / nP AS decimal(6, 0))
AS quantitaIng, i.unitaMisura
FROM ingrediente i, alimento a
WHERE i.codiceRicetta = codRicetta AND a.codiceALi = i.codiceAli;
END

```

Output della procedura inserendo nPersone='20' e codRicetta='S0002'

nomeALi	quantitaIng	unitaMisura
PESCE BACCALA' WM 60/80	1000	GR
CAPPERI ACETO VASO KG 0,72 NR=KG 0,72	20	GR
LATTE UHT P.S. LT 1 NR=1LT	1600	GR
PASTA D'ACCIUGHE KG 1 NR=KG 1	20	GR
CIPOLLE BIANCHE NR=CASSETTA DA 120	200	GR
OLIO EXTRA VERGINE OLIVA LT 5 NR=5LT	600	GR
FORMAGGIO GRANA GRATTUGIATO NR=1KG	100	GR
PREZZEMOLO SURGELATO KG 1 NR=KG 1	60	GR

2. Procedura che permette di ricercare il codice e la categoria di un articolo, ricevendo in input il nome dell'articolo o parte di esso.

```

CREATE PROCEDURE cheCodiceHaQuestoArticolo (IN nomeIn VARCHAR(50))
BEGIN
SELECT codice, nome, categoria
FROM (
    SELECT codiceArtMag AS codice, nomeArt AS nome, tipoArt AS
    categoria
    FROM articolomag
    UNION ALL
    SELECT codiceAli AS codice, nomeAli AS nome, 'alimenti' AS
    categoria
    FROM alimento
) AS unione
WHERE nome LIKE CONCAT('%', nomeIn, '%');
END

```

Output della procedura inserendo nomeIn='arot'

codice	nome	categoria
A0010	CAROTA NR=CASSETTA DA 160	alimenti
A0019	CAROTE RONDELLE GELO KG 2,5	alimenti

TRIGGER

1. I seguenti trigger permettono che il campo codice di articolo, alimento e ricetta siano sempre codificati con una stringa del seguente tipo: lettera che indica la categoria dell'articolo/ricetta seguita da un valore numerico. Mentre il codice identificativo delle richieste è costituito solamente da un numero, che però deve progredire automaticamente, dev'essere come se richiestaAlimento e richiestaArtMag fossero la stessa entità e quindi come se avessero la stessa chiave primaria.

```
CREATE TRIGGER autoAlimento BEFORE INSERT ON alimento
FOR EACH ROW
BEGIN
    SET @num = (
        SELECT sCounter
        FROM contatori
        WHERE tipo = 'alimenti'
    );
    SET @num = @num + 1;
    UPDATE contatori SET sCounter = @num WHERE tipo = 'alimenti';
    SET NEW.codiceAli = concat('A', REPEAT('0', 4 - LENGTH(@num)),
    @num);
END
```

```
CREATE TRIGGER autoArticolo BEFORE INSERT ON articolomag
FOR EACH ROW
BEGIN
    SET @num = (
        SELECT sCounter
        FROM contatori
        WHERE tipo = new.tipoArt
    );
    SET @num = @num + 1;
    UPDATE contatori SET sCounter = @num WHERE tipo = new.tipoArt;
    SET NEW.codiceArtMag = concat(UPPER(LEFT(new.tipoArt, 1)),
    REPEAT('0', 4 - LENGTH(@num)), @num);
END
```

```
CREATE TRIGGER autoRicetta BEFORE INSERT ON ricetta
FOR EACH ROW
BEGIN
    SET @num = (
        SELECT sCounter
        FROM contatori
        WHERE tipo = new.portata
    );
    SET @num = @num + 1;
    UPDATE contatori SET sCounter = @num WHERE tipo = new.portata;
    SET NEW.codiceRicetta = concat(UPPER(LEFT(new.portata, 1)),
    REPEAT('0', 4 - LENGTH(@num)), @num);
END
```

```

CREATE TRIGGER autoRichiestaAli BEFORE INSERT ON richiestaalimento
FOR EACH ROW
BEGIN
    SET @numTot = (
        SELECT COUNT(*)
        FROM (
            SELECT codiceRic
            FROM richiestaalimento
            UNION ALL
            SELECT codiceRic
            FROM richiestaartmag
        ) AS unione
    );
    SET @numTot = @numTot + 1;
    SET new.codiceRic = @numTot;
END

```

```

CREATE TRIGGER autoRichiestaArt BEFORE INSERT ON richiestaartmag
FOR EACH ROW
BEGIN
    SET @numTot = (
        SELECT COUNT(*)
        FROM (
            SELECT codiceRic
            FROM richiestaalimento
            UNION ALL
            SELECT codiceRic
            FROM richiestaartmag
        ) AS unione
    );
    SET @numTot = @numTot + 1;
    SET new.codiceRic = @numTot;
END

```

2. Trigger per eseguire un controllo sulla quantità di un articolo specificata in una richiesta, questa non può essere superiore a quella presente in magazzino. Il trigger è valido se nella richiesta si specifica la quantità con la stessa unità di misura usata nella tabella del magazzino.

I trigger di controllo sulla quantità richiesta sono stati aggiunti ai trigger autoRichiestaArt e autoRichiestaAli perché alcune versioni di SQL potrebbero non supportare il fatto che una tabella possieda due trigger dello stesso tipo (ovvero before insert) il risultato è il seguente:

```
CREATE TRIGGER autoRichiestaAli BEFORE INSERT ON richiestaalimento
FOR EACH ROW
BEGIN
SET @numTot = (
    SELECT COUNT(*)
    FROM (

        SELECT codiceRic
        FROM richiestaalimento
        UNION ALL
        SELECT codiceRic
        FROM richiestartaartmag
    ) AS unione
);
SET @numTot = @numTot + 1;
SET new.codiceRic = @numTot;

-- trigger controlloQuantita
SELECT a.quantitaAli, a.unitaMisura INTO @q, @u
FROM alimento a
WHERE a.codiceAli = new.codiceAli;
IF(@u = new.unitaMisura) THEN
    IF(@q < new.quantitaRic) THEN
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = "La quantita' richiesta e' superiore alla
        giacenza in magazzino";
    END IF;
ELSE
    IF(@u <> '') THEN
        SET @message = CONCAT("L'unita' di misura da inserire deve
        essere: ", @u);
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = @message;
    ELSE
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = "L'unita' di misura non e' stata
        specificata in magazzino, lasciare il campo vuoto (se c'e'
        levare la spunta da NULL)";
    END IF;
END IF;
END IF;
END
```

```

CREATE TRIGGER autoRichiestaArt BEFORE INSERT ON richiestartmag
FOR EACH ROW
BEGIN
SET @numTot = (
    SELECT COUNT(*)
    FROM (

        SELECT codiceRic
        FROM richiestaalimento
        UNION ALL
        SELECT codiceRic
        FROM richiestartmag
    ) AS unione
);
SET @numTot = @numTot + 1;
SET new.codiceRic = @numTot;

-- trigger controlloQuantita
SELECT a.quantitaArt, a.unitaMisura INTO @q, @u
FROM articolomag a
WHERE a.codiceArtMag = new.codiceArtMag;
IF(@u = new.unitaMisura) THEN
    IF(@q < new.quantitaRic) THEN
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = "La quantita' richiesta e' superiore alla
giacenza in magazzino";
    END IF;
ELSE
    IF(@u <> '') THEN
        SET @message = CONCAT("L'unita' di misura da inserire deve
essere: ", @u);
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = @message;
    ELSE
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = "L'unita' di misura non e' stata
specificata in magazzino, lasciare il campo vuoto (se c'e'
levare la spunta da NULL)";
    END IF;
END IF;
END

```

FUNZIONI

1. Funzione che controlli se un alimento è scaduto dato il codice di quell'alimento

```
CREATE FUNCTION controllaScaduti (codiceA VARCHAR(15)) RETURNS
VARCHAR(80)
BEGIN
    DECLARE scaduto BOOLEAN;
    DECLARE nome VARCHAR(80);
    SELECT COUNT(*) INTO scaduto FROM alimento
    WHERE codiceAli = codiceA AND dataScad < CURDATE();
    SELECT nomeAli INTO nome FROM alimento
    WHERE codiceAli = codiceA;
    IF(scaduto)
    THEN
        RETURN concat(nome, " È SCADUTO");
    ELSE
        RETURN concat(nome, " NON È SCADUTO");
    END IF;
END
```

2. Funzione che ritorni quanti alimenti sono scaduti

```
CREATE FUNCTION contaScaduti() RETURNS INT
BEGIN
    DECLARE contatore INT;
    SET contatore = (
        SELECT COUNT(*)
        FROM alimento
        WHERE dataScad < CURDATE()
    );
    RETURN contatore;
END
```

3. Funzione che verifica il nome di un articolo da richiedere a partire dal codice dell'articolo

```
CREATE FUNCTION nomeGiusto (codiceIn VARCHAR(15)) RETURNS VARCHAR(50)
BEGIN
    DECLARE nomeRis VARCHAR(50);
    SELECT nome INTO nomeRis
    FROM (
        SELECT codiceArtMag AS codice, nomeArt AS nome
        FROM articolomag
        UNION ALL
        SELECT codiceAli AS codice, nomeAli AS nome
        FROM alimento
    ) AS unione
    WHERE codice = codiceIn;
    RETURN nomeRis;
END
```