



Università degli Studi di Ferrara
Dipartimento di Ingegneria

Corso di Laurea Triennale in Ingegneria Elettronica
e Informatica

Documentazione Basi di Dati

Progetto E-Commerce: BINS

Studente:
Michele Vaccari
Matricola 121955

Docente:
Prof. Denis Ferraretti

Anno Accademico 2016–2017

INDICE

1	SPECIFICHE PROGETTO	1
1.1	Caratteristiche del sistema	1
1.2	Utenti del sistema	2
2	DIAGRAMMA E-R	3
2.1	Descrizione delle entità	3
2.1.1	Prodotto	3
2.1.2	Utente Amministratore	4
2.1.3	Utente Registrato	4
2.1.4	Ordine	4
2.1.5	Pagamento	4
2.1.6	Fattura	4
2.1.7	Ricetta	4
2.1.8	Marchio	4
2.1.9	Caratteristica	4
2.1.10	Reparto	4
2.1.11	Coupon	4
2.2	Descrizione delle associazioni	4
2.3	Il modello E-R completo	4
3	MODELLO RELAZIONALE	6
3.1	L'algoritmo di traduzione	6
3.1.1	Traduzione di entità	6
3.1.2	Traduzione di entità deboli	6
3.1.3	Traduzione di associazioni binarie 1:1	6
3.1.4	Traduzione di associazioni binarie 1:N	7
3.1.5	Traduzione di associazioni binarie N:M	7
3.2	Applicazione dell'algoritmo di traduzione	7
4	NORMALIZZAZIONE	13
4.1	Elenco entità	13
5	CODICE SQL	14
5.1	DDL	14
5.1.1	UTENTE_AMMINISTRATORE	14
5.1.2	UTENTE_REGISTRATO	14
5.1.3	FATTURA	15
5.1.4	PAGAMENTO	15
5.1.5	ORDINE	15
5.1.6	TRACKING	16
5.1.7	INDIRIZZO_DI_SPEDIZIONE	16
5.1.8	LISTA_DELLA_SPESA	16

5.1.9	CARRELLO	17
5.1.10	INDIRIZZO	17
5.1.11	RICETTA	17
5.1.12	MARCHIO	18
5.1.13	CARATTERISTICA	18
5.1.14	REPARTO	18
5.1.15	COUPON	19
5.1.16	PRODOTTO	19
5.1.17	SCADENZA	19
5.1.18	HA_PARTICOLARE	20
5.1.19	IMPIEGATO_IN	20
5.1.20	CONTENUTO_IN_LISTA	20
5.1.21	CONTENUTO_IN_CARRELLO	21
5.1.22	COMPONE	21
5.1.23	GESTISCE_PRODOTTI	21
5.2	Interrogazioni	21
6	INTERFACCIA GRAFICA	22
6.1	Elenco entità	22

ELENCO DELLE FIGURE

Figura 1	Entità Prodotto	3	
Figura 2	Schema E-R completo	5	
Figura 3	Traduzione di entità	8	
Figura 4	Traduzione di entità deboli	9	
Figura 5	Traduzione di associazioni binarie 1:1	10	
Figura 6	Traduzione di associazioni binarie 1:N	11	
Figura 7	Traduzione di associazioni binarie M:N	12	

ELENCO DELLE TABELLE

INTRODUZIONE

BINS (acronimo ricorsivo di "BINS Is Not Shopping") è un'applicazione web liberamente disponibile, indicata soprattutto per commercializzare prodotti alimentari. Lo scopo di questo lavoro è di documentare l'applicazione usando l'UML (Unified Model Language).

La documentazione è articolata come segue.

IL PRIMO CAPITOLO contiene le specifiche del progetto descrivendo le caratteristiche e le categorie di utenti dell'applicazione.

IL SECONDO CAPITOLO spiega le operazioni, veramente semplici, per installare \LaTeX sul proprio calcolatore.

IL TERZO CAPITOLO spiega le operazioni, veramente semplici, per installare \LaTeX sul proprio calcolatore.

IL QUARTO CAPITOLO spiega le operazioni, veramente semplici, per installare \LaTeX sul proprio calcolatore.

IL QUINTO CAPITOLO spiega le operazioni, veramente semplici, per installare \LaTeX sul proprio calcolatore.

IL SESTO CAPITOLO spiega le operazioni, veramente semplici, per installare \LaTeX sul proprio calcolatore.

L'implementazione dell'applicazione è stata fatta utilizzando il framework *Bootstrap* per ingegnerizzare al meglio il livello di presentazione.

Per l'application server si è usato *Tomcat* e si è utilizzato il linguaggio di programmazione Java. Per interfacciare l'application server con il database si è utilizzato JDBC.

Il DBMS utilizzato è *MySQL*.

1.1 CARATTERISTICHE DEL SISTEMA

Si vuole progettare un'applicazione web per la vendita on line dei prodotti di un supermercato. L'applicazione deve avere un frontend multilingue (italiano e inglese).

I prodotti del supermercato sono suddivisi in reparti (ortofrutta, macelleria, accessori per la casa, cura della persona, ecc.) e ogni prodotto può avere una data di scadenza.

L'applicazione web deve avere le seguenti caratteristiche:

1. Possibilità di visualizzare il catalogo dei prodotti, navigabile per reparto, caratteristiche, marchio, data di scadenza, ricerca libera, ecc. Possibilità di vedere il singolo prodotto con tutti i dettagli;
2. Possibilità di inserire prodotti nel carrello e di effettuare l'acquisto di più prodotti in diverse quantità. Predisporre la gestione dei prezzi, del totale carrello, la gestione della disponibilità di magazzino, impedendo di poter acquistare quantità non disponibili e prodotti scaduti;
3. Gestione di una o più shopping list (lista della spesa);
4. Possibilità di inserire o selezionare un indirizzo di consegna da una rubrica personale, di simulare il pagamento e di confermare l'ordine;
5. Possibilità di visualizzare lo stato dell'ordine e lo storico degli ordini effettuati;
6. Gestione di coupon o buoni sconto in fase di acquisto, possibilità di visualizzare il tracking dell'ordine in consegna;
7. Possibilità di gestire il catalogo (inserimento, modifica, blocco prodotti), i reparti, i brand, ecc. Possibilità di gestire la disponibilità di magazzino;
8. Gestione di prodotti in push (per cui spingere la vendita) con inserimento in una "vetrina in home page" o in un'area promo;
9. Possibilità di visualizzare gli utenti, verificare il numero degli ordini per utente, bloccare eventuali utenti, gestire gli altri utenti amministratori;
10. Gestione dei coupon o buoni sconto;

11. Gestione del tracking dell'ordine, con simulazione di tutti i cambi di stato (consegnato al corriere, in viaggio, consegnato al destinatario, ecc.);

1.2 UTENTI DEL SISTEMA

Il sistema prevede che le categorie di utenti sia così rappresentata:

UTENTI PUBBLICI Possono effettuare solo il punto [1](#) ed eventualmente registrarsi.

UTENTI REGISTRATI Possono effettuare i punti dal [1](#) al [6](#) compresi.

AMMINISTRATORI Possono effettuare i punti dal [7](#) al [11](#) compresi.

2

DIAGRAMMA E-R

Una volta analizzate le specifiche del progetto per il quale si vuole realizzare il database, si identificano e si descrivono le singole entità e le loro associazioni.

2.1 DESCRIZIONE DELLE ENTITÀ

bla bla bla...

2.1.1 Prodotto

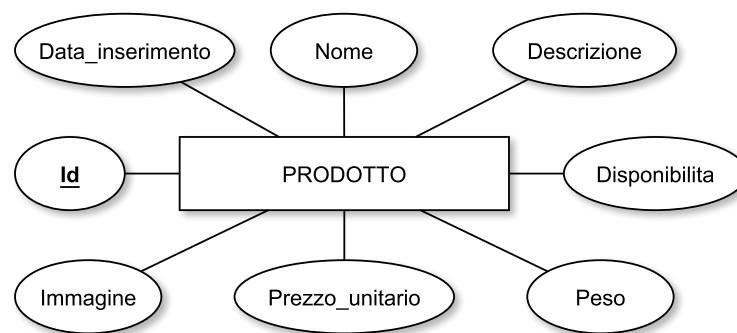


Figura 1: Entità Prodotto

Descrizione Attributi

ID Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

DATA INSERIMENTO Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

NOME Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

DESCRIZIONE Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

DISPONIBILITA Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

PESO Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

PREZZO UNITARIO Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

IMMAGINE Chiave primaria dell'entità che identifica univocamente il prodotto. Il valore di tale attributo viene generato mediante un contatore.

2.1.2 Utente Amministratore

2.1.3 Utente Registrato

2.1.4 Ordine

2.1.5 Pagamento

2.1.6 Fattura

2.1.7 Ricetta

2.1.8 Marchio

2.1.9 Caratteristica

2.1.10 Reparto

2.1.11 Coupon

2.2 DESCRIZIONE DELLE ASSOCIAZIONI

2.3 IL MODELLO E-R COMPLETO

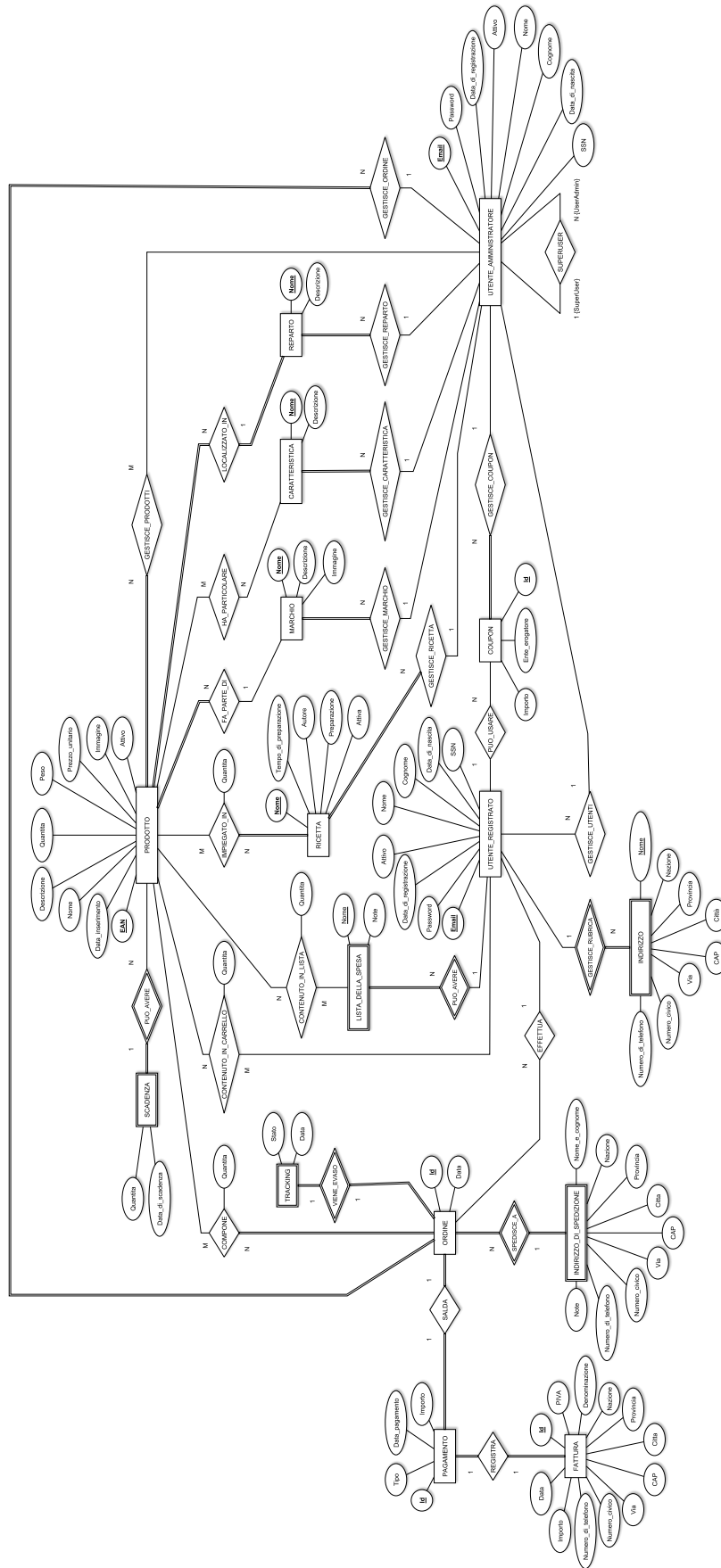


Figura 2: Schema E-R completo

3 | MODELLO RELAZIONALE

3.1 L'ALGORITMO DI TRADUZIONE

La fase successiva della progettazione di una base di dati consiste nel passare dalla progettazione concettuale alla progettazione logica. Per passare al modello relazionale, si applica un semplice algoritmo.

L'algoritmo che si utilizza per la traduzione dal modello E-R in modello relazionale è formato dalle seguenti operazioni:

1. Traduzione di tipi di entità;
2. Traduzione di tipi di entità deboli;
3. Traduzione di associazioni binarie di tipo 1 : 1;
4. Traduzione di associazioni binarie di tipo 1 : N;
5. Traduzione di associazioni binarie di tipo N : M;

3.1.1 Traduzione di entità

Per ogni tipo di entità (forte) nello schema E-R si costruisce una relazione che contiene tutti gli attributi semplici dell'entità.

3.1.2 Traduzione di entità deboli

Per ogni tipo di entità debole dello schema E-R con tipo di entità proprietario, si costruisce una relazione e si inseriscono tutti gli attributi semplici dell'entità debole come attributi della relazione. Si inseriscono come attributi di chiave esterna della relazione gli attributi di chiave primaria delle relazioni corrispondenti ai tipi di entità proprietari. La chiave primaria della relazione è data dalla combinazione delle chiavi primarie delle entità proprietarie e dalla chiave parziale del tipo di entità debole (se esiste).

3.1.3 Traduzione di associazioni binarie 1:1

Per ogni tipo di associazione binaria 1 : 1 nello schema E-R si individuano le due relazioni coinvolte dall'associazione. Per la traduzione si usa l'approccio basato su chiavi esterne. Si sceglie una delle due relazioni coinvolte, preferibilmente la relazione corrispondente a un tipo di entità con vincolo di partecipazione totale, e si inserisce come chiave esterna la chiave primaria

della seconda relazione. Infine si inseriscono sulla relazione scelta tutti gli attributi semplici del tipo di associazione 1 : 1.

3.1.4 Traduzione di associazioni binarie 1:N

Per ogni tipo di associazione binaria 1 : N nello schema E-R si individuano le due relazioni coinvolte dall'associazione. Per la traduzione si individua la relazione che rappresenta il tipo di entità partecipante lato-N del tipo di associazione e si inserisce come chiave esterna la chiave primaria della relazione che rappresenta l'altro tipo di entità partecipante all'associazione. Infine si inseriscono sulla relazione scelta tutti gli attributi semplici del tipo di associazione 1 : N.

3.1.5 Traduzione di associazioni binarie N:M

Per ogni tipo di associazione binaria M : N nello schema E-R si costruisce una nuova relazione che rappresenta l'associazione. Si inseriscono come attributi di chiave esterna della relazione le chiavi primarie delle relazioni che rappresentano i tipi di entità partecipanti, le loro combinazioni formano la chiave primaria della relazione. Infine si inseriscono nella nuova relazione tutti gli attributi semplici del tipo di associazione M : N.

3.2 APPLICAZIONE DELL'ALGORITMO DI TRADUZIONE

Di seguito si riporta la traduzione in modello relazionale del modello E-R analizzato:

PRODOTTO

<u>EAN</u>	Data_inserimento	Nome	Descrizione	Quantita	Peso	Prezzo_unitario	Immagine	Attivo
------------	------------------	------	-------------	----------	------	-----------------	----------	--------

RICETTA

<u>Nome</u>	<u>Autore</u>	Tempo_di_preparazione	Preparazione	Attiva
-------------	---------------	-----------------------	--------------	--------

MARCHIO

<u>Nome</u>	Descrizione	Immagine
-------------	-------------	----------

CARATTERISTICA

<u>Nome</u>	Descrizione
-------------	-------------

REPARTO

<u>Nome</u>	Descrizione
-------------	-------------

UTENTE_AMMINISTRATORE

<u>Email</u>	Password	Data_di_registrazione	Attivo	Nome	Cognome	Data_di_nascita	SSN
--------------	----------	-----------------------	--------	------	---------	-----------------	-----

COUPON

<u>Id</u>	Ente_erogatore	Importo
-----------	----------------	---------

UTENTE_REGISTRATO

<u>Email</u>	Password	Data_di_registrazione	Attivo	Nome	Cognome	Data_di_nascita	SSN
--------------	----------	-----------------------	--------	------	---------	-----------------	-----

ORDINE

<u>Id</u>	Data
-----------	------

PAGAMENTO

<u>Id</u>	Tipo	Data_pagamento	Importo
-----------	------	----------------	---------

FATTURA

<u>Id</u>	PIVA	Denominazione	Nazione	Provincia	Citta	CAP	Via	Numero_civico	Numero_di_telefono	Importo	Data
-----------	------	---------------	---------	-----------	-------	-----	-----	---------------	--------------------	---------	------

Figura 3: Traduzione di entità

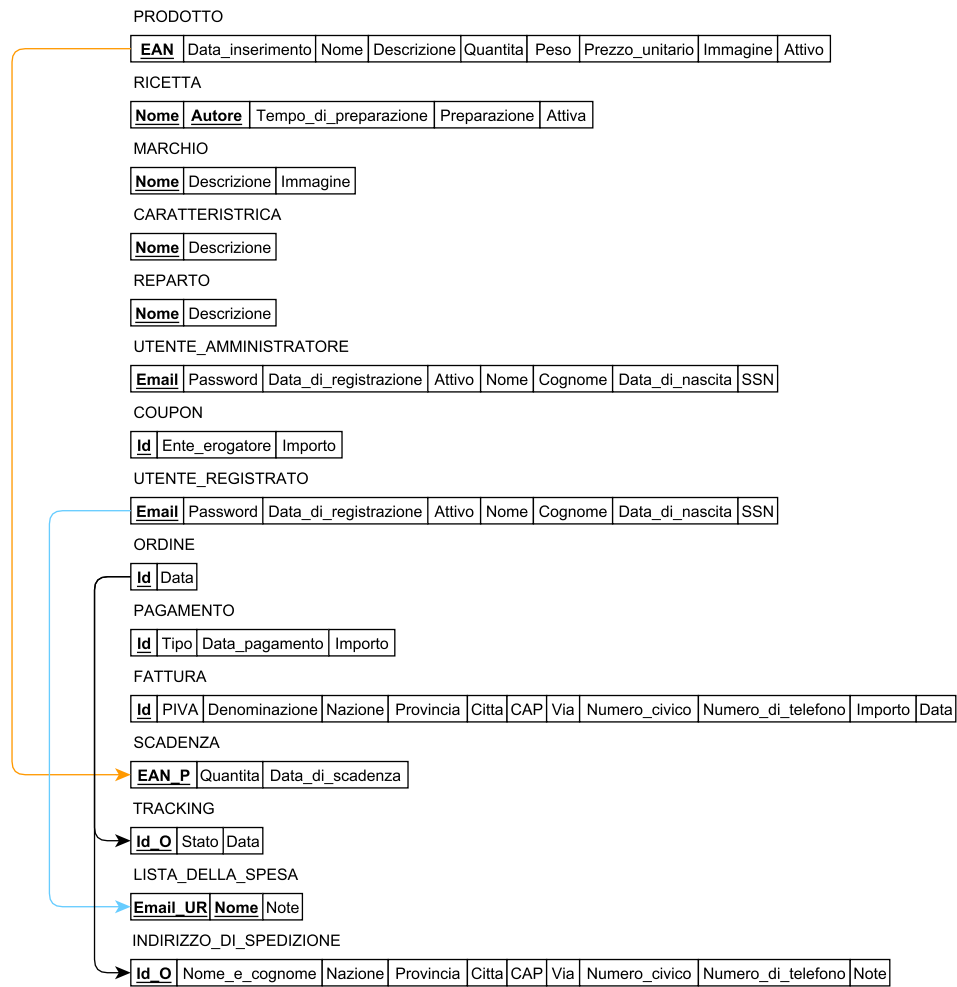


Figura 4: Traduzione di entità deboli

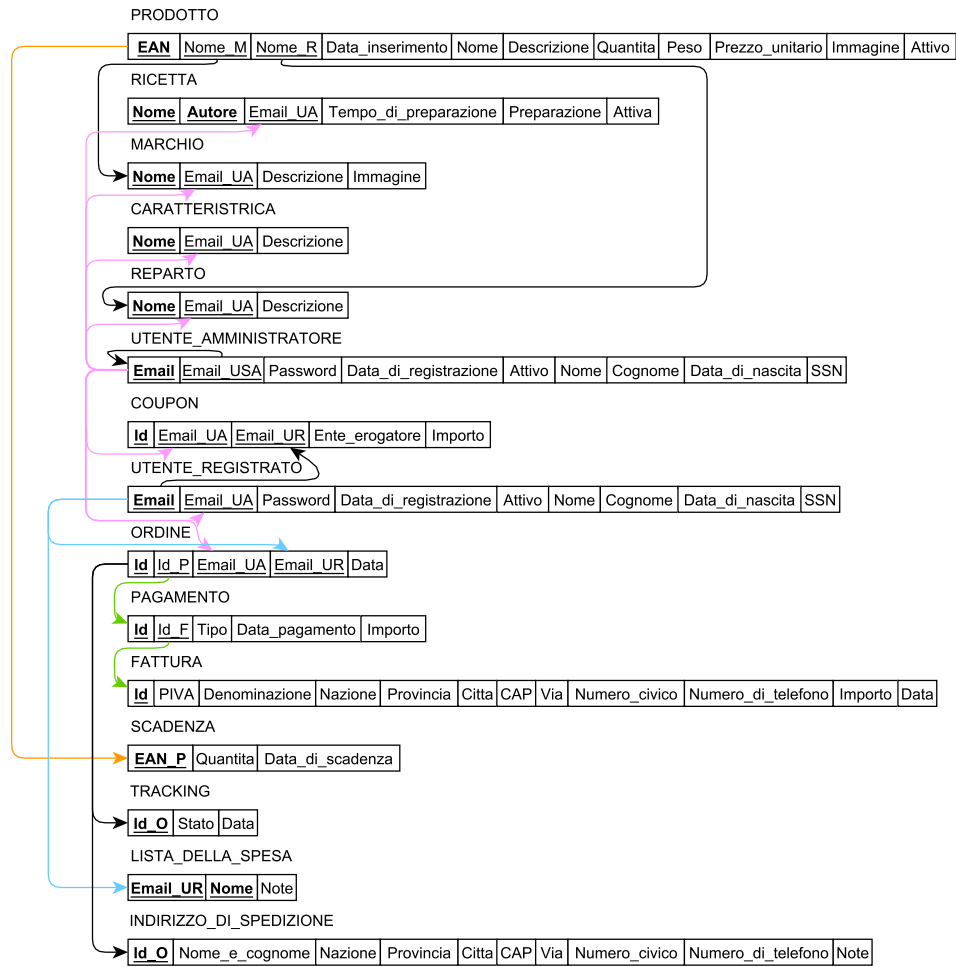


Figura 6: Traduzione di associazioni binarie 1:N

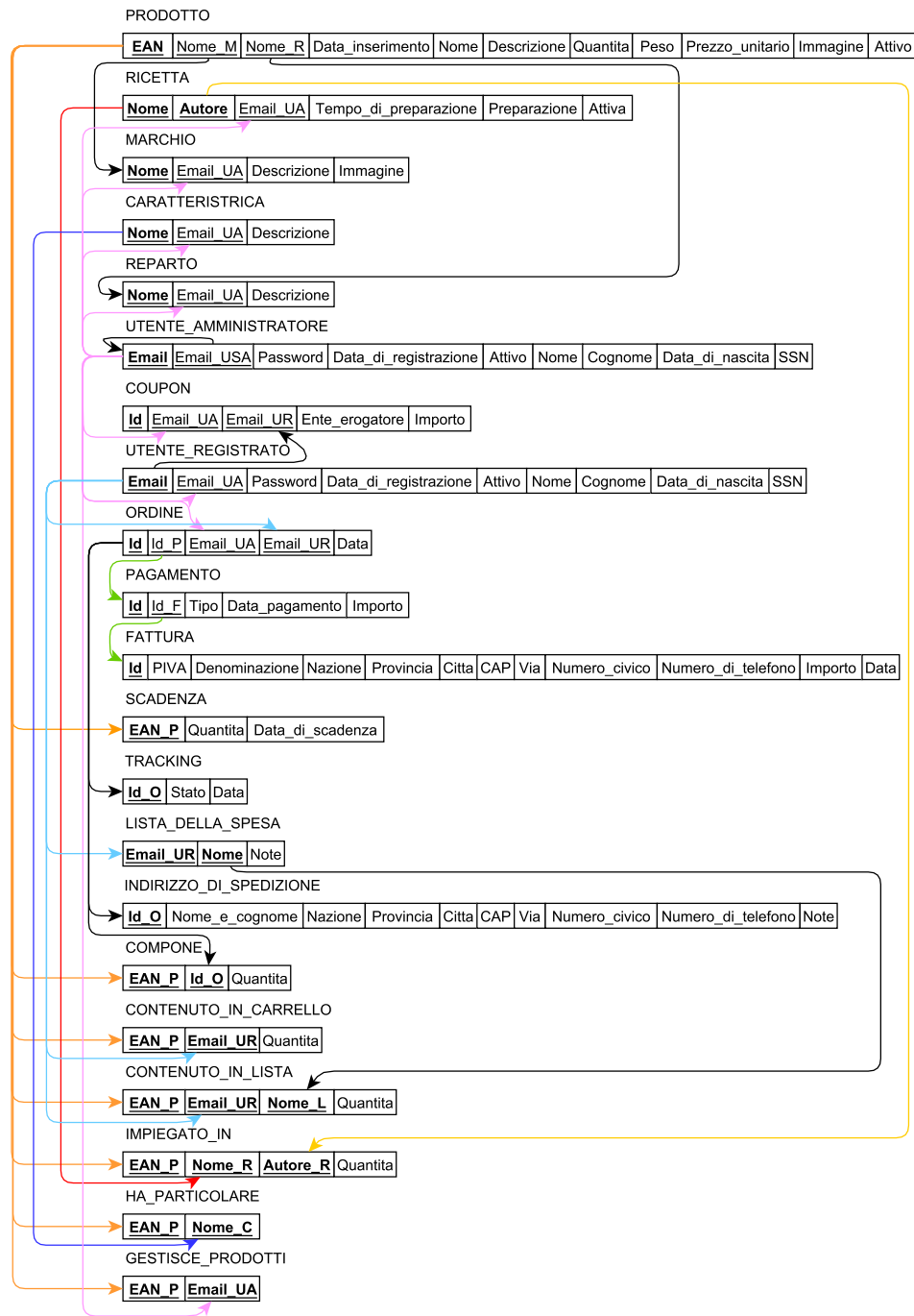


Figura 7: Traduzione di associazioni binarie M:N

4 | NORMALIZZAZIONE

4.1 ELENCO ENTITÀ

bla bla bla...

5

CODICE SQL

5.1 DDL

Si riporta il codice SQL (estrapolato da phpMyAdmin) utilizzato per creare le tabelle che costituiscono la base di dati e i vincoli di integrità referenziale.

5.1.1 UTENTE_AMMINISTRATORE

Struttura della tabella *UTENTE_AMMINISTRATORE*:

```
CREATE TABLE IF NOT EXISTS 'UTENTE_AMMINISTRATORE' (  
    'Email' VARCHAR(100) NOT NULL,  
    'Email_UAS' VARCHAR(100) NOT NULL DEFAULT 'admin@bins.com',  
    'Password' VARCHAR(32) NOT NULL,  
    'Data_di_registrazione' TIMESTAMP NOT NULL,  
    'Attivo' BOOLEAN NOT NULL DEFAULT TRUE,  
    'Nome' VARCHAR(50) NOT NULL,  
    'Cognome' VARCHAR(50) NOT NULL,  
    'Data_di_nascita' DATE NOT NULL,  
    'SSN' VARCHAR(16) NOT NULL,  
    PRIMARY KEY ('Email'),  
    FOREIGN KEY ('Email_UAS') REFERENCES UTENTE_AMMINISTRATORE('Email')  
) ENGINE = InnoDB;
```

5.1.2 UTENTE_REGISTRATO

Struttura della tabella *UTENTE_REGISTRATO*:

```
CREATE TABLE IF NOT EXISTS 'UTENTE_REGISTRATO' (  
    'Email' VARCHAR(100) NOT NULL,  
    'Email_UA' VARCHAR(100) NOT NULL DEFAULT 'admin@bins.com',  
    'Password' VARCHAR(32) NOT NULL,  
    'Data_di_registrazione' TIMESTAMP NOT NULL,  
    'Attivo' BOOLEAN NOT NULL DEFAULT TRUE,  
    'Nome' VARCHAR(50) NOT NULL,  
    'Cognome' VARCHAR(50) NOT NULL,  
    'Data_di_nascita' DATE NOT NULL,  
    'SSN' VARCHAR(16) NOT NULL,  
    PRIMARY KEY ('Email'),  
    FOREIGN KEY ('Email_UA') REFERENCES UTENTE_AMMINISTRATORE('Email')  
) ENGINE = InnoDB;
```

5.1.3 FATTURA

Struttura della tabella *FATTURA*:

```
CREATE TABLE IF NOT EXISTS 'FATTURA' (
  'Id' INT UNSIGNED NOT NULL AUTO_INCREMENT,
  'PIVA' VARCHAR(16) NOT NULL,
  'Denominazione' VARCHAR(100) NOT NULL,
  'Nazione' VARCHAR(50) NOT NULL,
  'Provincia' VARCHAR(50) NOT NULL,
  'Citta' VARCHAR(50) NOT NULL,
  'CAP' VARCHAR(5) NOT NULL,
  'Via' VARCHAR(100) NOT NULL,
  'Numero_civico' VARCHAR(10) NOT NULL,
  'Numero_di_telefono' VARCHAR(30) NOT NULL,
  'Importo' DOUBLE(8,2) UNSIGNED NOT NULL DEFAULT '0.00',
  'Data' TIMESTAMP NOT NULL,
  PRIMARY KEY ('Id')
) ENGINE = InnoDB AUTO_INCREMENT=1;
```

5.1.4 PAGAMENTO

Struttura della tabella *PAGAMENTO*:

```
CREATE TABLE IF NOT EXISTS 'PAGAMENTO' (
  'Id' INT UNSIGNED NOT NULL AUTO_INCREMENT,
  'Id_F' INT UNSIGNED NOT NULL,
  'Data_pagamento' TIMESTAMP NOT NULL,
  'Importo' DOUBLE(8,2) UNSIGNED NOT NULL DEFAULT '0.00',
  PRIMARY KEY ('Id'),
  FOREIGN KEY ('Id_F') REFERENCES FATTURA('Id')
) ENGINE = InnoDB AUTO_INCREMENT=1;
```

5.1.5 ORDINE

Struttura della tabella *ORDINE*:

```
CREATE TABLE IF NOT EXISTS 'ORDINE' (
  'Id' INT UNSIGNED NOT NULL AUTO_INCREMENT,
  'Email_UA' VARCHAR(100) NOT NULL,
  'Email_UR' VARCHAR(100) NOT NULL,
  'Id_P' INT UNSIGNED NOT NULL,
  'Data' TIMESTAMP NOT NULL,
  PRIMARY KEY ('Id'),
  FOREIGN KEY ('Email_UA') REFERENCES UTENTE_AMMINISTRATORE('Email'),
  FOREIGN KEY ('Email_UR') REFERENCES UTENTE_REGISTRATO('Email'),
  FOREIGN KEY ('Id_P') REFERENCES PAGAMENTO('Id')
) ENGINE = InnoDB AUTO_INCREMENT=1;
```

5.1.6 TRACKING

Struttura della tabella *TRACKING*:

```
CREATE TABLE IF NOT EXISTS 'TRACKING' (
  'Id_0' INT UNSIGNED NOT NULL,
  'Stato' VARCHAR(100) NOT NULL,
  'Data' TIMESTAMP NOT NULL,
  PRIMARY KEY ('Id_0'),
  FOREIGN KEY ('Id_0') REFERENCES ORDINE('Id')
) ENGINE = InnoDB;
```

5.1.7 INDIRIZZO_DI_SPEDIZIONE

Struttura della tabella *INDIRIZZO_DI_SPEDIZIONE*:

```
CREATE TABLE IF NOT EXISTS 'INDIRIZZO_DI_SPEDIZIONE' (
  'Id_0' INT UNSIGNED NOT NULL,
  'Tipo' VARCHAR(20) NOT NULL,
  'Costo' DOUBLE(8,2) UNSIGNED NOT NULL DEFAULT '0.00',
  'Nome_e_cognome' VARCHAR(100) NOT NULL,
  'Nazione' VARCHAR(50) NOT NULL,
  'Provincia' VARCHAR(50) NOT NULL,
  'Citta' VARCHAR(50) NOT NULL,
  'CAP' VARCHAR(5) NOT NULL,
  'Via' VARCHAR(100) NOT NULL,
  'Numero_civico' VARCHAR(10) NOT NULL,
  'Numero_di_telefono' VARCHAR(30) NOT NULL,
  'Note' TEXT(1000),
  PRIMARY KEY ('Id_0'),
  FOREIGN KEY ('Id_0') REFERENCES ORDINE('Id')
) ENGINE = InnoDB;
```

5.1.8 LISTA_DELLA_SPESA

Struttura della tabella *LISTA_DELLA_SPESA*:

```
CREATE TABLE IF NOT EXISTS 'LISTA_DELLA_SPESA' (
  'Email_UR' VARCHAR(100) NOT NULL,
  'Nome' VARCHAR(50) NOT NULL,
  PRIMARY KEY ('Email_UR', 'Nome'),
  FOREIGN KEY ('Email_UR') REFERENCES UTENTE_REGISTRATO('Email')
) ENGINE = InnoDB;
```

5.1.9 CARRELLO

Struttura della tabella *CARRELLO*:

```
CREATE TABLE IF NOT EXISTS 'CARRELLO' (
    'Email_UR' VARCHAR(100) NOT NULL,
    'Totale_numero_prodotti' INT UNSIGNED NOT NULL,
    'Prezzo_Totale' DOUBLE(8,2) UNSIGNED NOT NULL DEFAULT '0.00',
    PRIMARY KEY ('Email_UR'),
    FOREIGN KEY ('Email_UR') REFERENCES UTENTE_REGISTRATO('Email')
) ENGINE = InnoDB;
```

5.1.10 INDIRIZZO

Struttura della tabella *INDIRIZZO*:

```
CREATE TABLE IF NOT EXISTS 'INDIRIZZO' (
    'Email_UR' VARCHAR(100) NOT NULL,
    'Nome' VARCHAR(50) NOT NULL,
    'Nazione' VARCHAR(50) NOT NULL,
    'Provincia' VARCHAR(50) NOT NULL,
    'Citta' VARCHAR(50) NOT NULL,
    'CAP' VARCHAR(5) NOT NULL,
    'Via' VARCHAR(100) NOT NULL,
    'Numero_civico' VARCHAR(10) NOT NULL,
    'Numero_di_telefono' VARCHAR(30) NOT NULL,
    PRIMARY KEY ('Email_UR', 'Nome'),
    FOREIGN KEY ('Email_UR') REFERENCES UTENTE_REGISTRATO('Email')
) ENGINE = InnoDB;
```

5.1.11 RICETTA

Struttura della tabella *RICETTA*:

```
CREATE TABLE IF NOT EXISTS 'RICETTA' (
    'Nome' VARCHAR(50) NOT NULL,
    'Autore' VARCHAR(50) NOT NULL,
    'Email_UA' VARCHAR(50) NOT NULL,
    'Tempo_di_preparazione' TIME NOT NULL,
    'Preparazione' TEXT(22000) NOT NULL,
    'Attiva' BOOLEAN NOT NULL DEFAULT TRUE,
    PRIMARY KEY ('Nome', 'Autore'),
    FOREIGN KEY ('Email_UA') REFERENCES UTENTE_AMMINISTRATORE('Email')
) ENGINE = InnoDB;
```

5.1.12 MARCHIO

Struttura della tabella *MARCHIO*:

```
CREATE TABLE IF NOT EXISTS 'MARCHIO' (  
    'Nome' VARCHAR(50) NOT NULL,  
    'Email-UA' VARCHAR(50) NOT NULL,  
    'Descrizione' TEXT(2000) NOT NULL,  
    'Immagine' VARCHAR(100) NOT NULL,  
    PRIMARY KEY ('Nome'),  
    FOREIGN KEY ('Email-UA') REFERENCES UTENTE_AMMINISTRATORE('Email  
        ')  
) ENGINE = InnoDB;
```

5.1.13 CARATTERISTICA

Struttura della tabella *CARATTERISTICA*:

```
CREATE TABLE IF NOT EXISTS 'CARATTERISTICA' (  
    'Nome' VARCHAR(50) NOT NULL,  
    'Email-UA' VARCHAR(50) NOT NULL,  
    'Descrizione' TEXT(2000) NOT NULL,  
    PRIMARY KEY ('Nome'),  
    FOREIGN KEY ('Email-UA') REFERENCES UTENTE_AMMINISTRATORE('Email  
        ')  
) ENGINE = InnoDB;
```

5.1.14 REPARTO

Struttura della tabella *REPARTO*:

```
CREATE TABLE IF NOT EXISTS 'REPARTO' (  
    'Nome' VARCHAR(50) NOT NULL,  
    'Email-UA' VARCHAR(50) NOT NULL,  
    'Descrizione' TEXT(2000) NOT NULL,  
    PRIMARY KEY ('Nome'),  
    FOREIGN KEY ('Email-UA') REFERENCES UTENTE_AMMINISTRATORE('Email  
        ')  
) ENGINE = InnoDB;
```


5.1.15 COUPON

Struttura della tabella *COUPON*:

```
CREATE TABLE IF NOT EXISTS 'COUPON' (
    'Id' INT UNSIGNED NOT NULL,
    'Email_UA' VARCHAR(50) NOT NULL,
    'Email_UR' VARCHAR(50) NOT NULL,
    'Ente_erogatore' VARCHAR(50) NOT NULL,
    'Importo' DOUBLE(8,2) NOT NULL DEFAULT '0.00',
    PRIMARY KEY ('Id'),
    FOREIGN KEY ('Email_UA') REFERENCES UTENTE_AMMINISTRATORE('Email'),
    FOREIGN KEY ('Email_UR') REFERENCES UTENTE_REGISTRATO('Email')
) ENGINE = InnoDB;
```

5.1.16 PRODOTTO

Struttura della tabella *PRODOTTO*:

```
CREATE TABLE IF NOT EXISTS 'PRODOTTO' (
    'EAN' VARCHAR(13) NOT NULL,
    'Nome_M' VARCHAR(50) NOT NULL,
    'Nome_R' VARCHAR(50) NOT NULL,
    'Data_inserimento' TIMESTAMP NOT NULL,
    'Attivo' BOOLEAN NOT NULL DEFAULT TRUE,
    'Nome' VARCHAR(100) NOT NULL,
    'Descrizione' TEXT(2000) NOT NULL,
    'Disponibilita' INT UNSIGNED NOT NULL,
    'Peso' INT UNSIGNED NOT NULL,
    'Prezzo_unitario' DOUBLE(8,2) UNSIGNED NOT NULL DEFAULT '0.00',
    'Immagine' VARCHAR(100) NOT NULL,
    PRIMARY KEY ('EAN'),
    FOREIGN KEY ('Nome_M') REFERENCES MARCHIO('Nome'),
    FOREIGN KEY ('Nome_R') REFERENCES RICETTA('Nome')
) ENGINE = InnoDB;
```

5.1.17 SCADENZA

Struttura della tabella *SCADENZA*:

```
CREATE TABLE IF NOT EXISTS 'SCADENZA' (
    'EAN_P' VARCHAR(13) NOT NULL,
    'Data_di_scadenza' DATE NOT NULL,
    'Quantita' INT UNSIGNED NOT NULL,
    PRIMARY KEY ('EAN_P'),
    FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN')
) ENGINE = InnoDB;
```

5.1.18 HA_PARTICOLARE

Struttura della tabella *HA_PARTICOLARE*:

```
CREATE TABLE IF NOT EXISTS 'HA_PARTICOLARE' (
    'EAN_P' VARCHAR(13) NOT NULL,
    'Nome_C' VARCHAR(50) NOT NULL,
    PRIMARY KEY ('EAN_P', 'Nome_C'),
    FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN'),
    FOREIGN KEY ('Nome_C') REFERENCES CARATTERISTICA('Nome')
) ENGINE = InnoDB;
```

5.1.19 IMPIEGATO_IN

Struttura della tabella *IMPIEGATO_IN*:

```
CREATE TABLE IF NOT EXISTS 'IMPIEGATO_IN' (
    'EAN_P' VARCHAR(13) NOT NULL,
    'Nome_R' VARCHAR(50) NOT NULL,
    'Autore_R' VARCHAR(100) NOT NULL,
    'Quantita' INT UNSIGNED NOT NULL,
    PRIMARY KEY ('EAN_P', 'Nome_R', 'Autore_R'),
    FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN'),
    FOREIGN KEY ('Nome_R', 'Autore_R') REFERENCES RICETTA('Nome', 'Autore')
) ENGINE = InnoDB;
```

5.1.20 CONTENUTO_IN_LISTA

Struttura della tabella *CONTENUTO_IN_LISTA*:

```
CREATE TABLE IF NOT EXISTS 'CONTENUTO_IN_LISTA' (
    'EAN_P' VARCHAR(13) NOT NULL,
    'Email_UR' VARCHAR(100) NOT NULL,
    'Nome_L' VARCHAR(50) NOT NULL,
    'Quantita' INT UNSIGNED NOT NULL,
    PRIMARY KEY ('EAN_P', 'Email_UR', 'Nome_L'),
    FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN'),
    FOREIGN KEY ('Email_UR', 'Nome_L') REFERENCES LISTA_DELLA_SPESA('Email_UR', 'Nome')
) ENGINE = InnoDB;
```

5.1.21 CONTENUTO_IN_CARRELLO

Struttura della tabella *CONTENUTO_IN_CARRELLO*:

```
CREATE TABLE IF NOT EXISTS 'CONTENUTO_IN_CARRELLO' (
  'EAN_P' VARCHAR(13) NOT NULL,
  'Email_UR' VARCHAR(100) NOT NULL,
  'Quantita' INT UNSIGNED NOT NULL,
  PRIMARY KEY ('EAN_P', 'Email_UR'),
  FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN'),
  FOREIGN KEY ('Email_UR') REFERENCES UTENTE_REGISTRATO('Email')
) ENGINE = InnoDB;
```

5.1.22 COMPONE

Struttura della tabella *COMPONE*:

```
CREATE TABLE IF NOT EXISTS 'COMPONE' (
  'EAN_P' VARCHAR(13) NOT NULL,
  'Id_0' INT UNSIGNED NOT NULL,
  'Quantita' INT UNSIGNED NOT NULL,
  PRIMARY KEY ('EAN_P', 'Id_0'),
  FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN'),
  FOREIGN KEY ('Id_0') REFERENCES ORDINE('Id')
) ENGINE = InnoDB;
```

5.1.23 GESTISCE_PRODOTTI

Struttura della tabella *GESTISCE_PRODOTTI*:

```
CREATE TABLE IF NOT EXISTS 'GESTISCE_PRODOTTI' (
  'EAN_P' VARCHAR(13) NOT NULL,
  'Email_UA' VARCHAR(100) NOT NULL,
  PRIMARY KEY ('EAN_P', 'Email_UA'),
  FOREIGN KEY ('EAN_P') REFERENCES PRODOTTO('EAN'),
  FOREIGN KEY ('Email_UA') REFERENCES UTENTE_AMMINISTRATORE('Email')
) ENGINE = InnoDB;
```

5.2 INTERROGAZIONI

Tra le molte interrogazioni utilizzate dall'applicazione web riguardo il database precedente si riportano quelle che meritano un commento.

6 | INTERFACCIA GRAFICA

6.1 ELENCO ENTITÀ

bla bla bla...