

Università degli Studi di Salerno

Corso di Ingegneria del Software



Sandwich on web

DBD

Versione 1.0

Data: 23/12/2015

Partecipanti:

Nome	Matricola
Sara Volpe	0512102434
Egidio Giacoia	0512102376
Nunzia Esposito	0512102328

Scritto da:	Team members
-------------	--------------

Revision History

Data	Versione	Descrizione	Autore
13/12/2015	1	Stesura del documento	Team members
15/12/2015	1	Aggiunta di: Modello ER	Egidio Giacoia Nunzia Esposito
16/12/2015	1	Aggiunta di: Dizionario dei dati, Modello EER ristrutturato	Nunzia Esposito
17/12/2015	1	Aggiunta di: Modello logico	Nunzia Esposito
19/12/2015	1	Creazione database	Egidio Giacoia Nunzia Esposito
20/12/2015	1	Modifica del codice di creazione del database: aggiunta dei trigger	Egidio Giacoia
21/12/2015	1	Aggiunta di: Design delle transazioni	Sara Volpe
22/12/2015	1	Revisione documento	Nunzia Esposito

INDICE

- 1. INTRODUZIONE
 - 1.1 Scelta del DBMS
- 2. PROGETTAZIONE DEL DATABASE
 - 2.1. Gestione dei dati persistenti
 - 2.1.1 Modello ER
 - 2.1.2 Modello ER ristrutturato
 - 2.1.3 Dizionario dei dati
 - 2.1.3.1 Entità
 - 2.1.3.2 Relazioni
 - 2.1.4 Modello logico
- 3. DESCRIZIONE DEL DATABASE
- 4. CODICE DI CREAZIONE DEL DATABASE
- 5. DESIGN DELLE TRANSAZIONI

1. Introduzione

Questo documento descrive nel dettaglio in DBMS che si è scelti di utilizzare e tutta la logica riguardante la gestione dei dati persistenti.

1.1 Scelta del DBMS

Il problema della persistenza dei dati nasce dalla necessità di rendere permanenti alcune informazioni all'interno di un sistema anche quando questo è spento. Molte volte capita di avere dei programmi con un gran numero di dati da gestire, dati che non possono essere reinseriti dopo un blackout o dopo lo spegnimento della macchina; si pensi ad un archivio di mille utenti contenente tutti i dati anagrafici. Sarebbe impensabile il

reinserimento degli stessi ogni volta che la macchina sul quale risiedono i dati si spegne. Bisogna, quindi, trovare il modo per poter tener traccia dei dati utili anche quando l'applicazione che li ha utilizzati e/o creati ha smesso di funzionare.

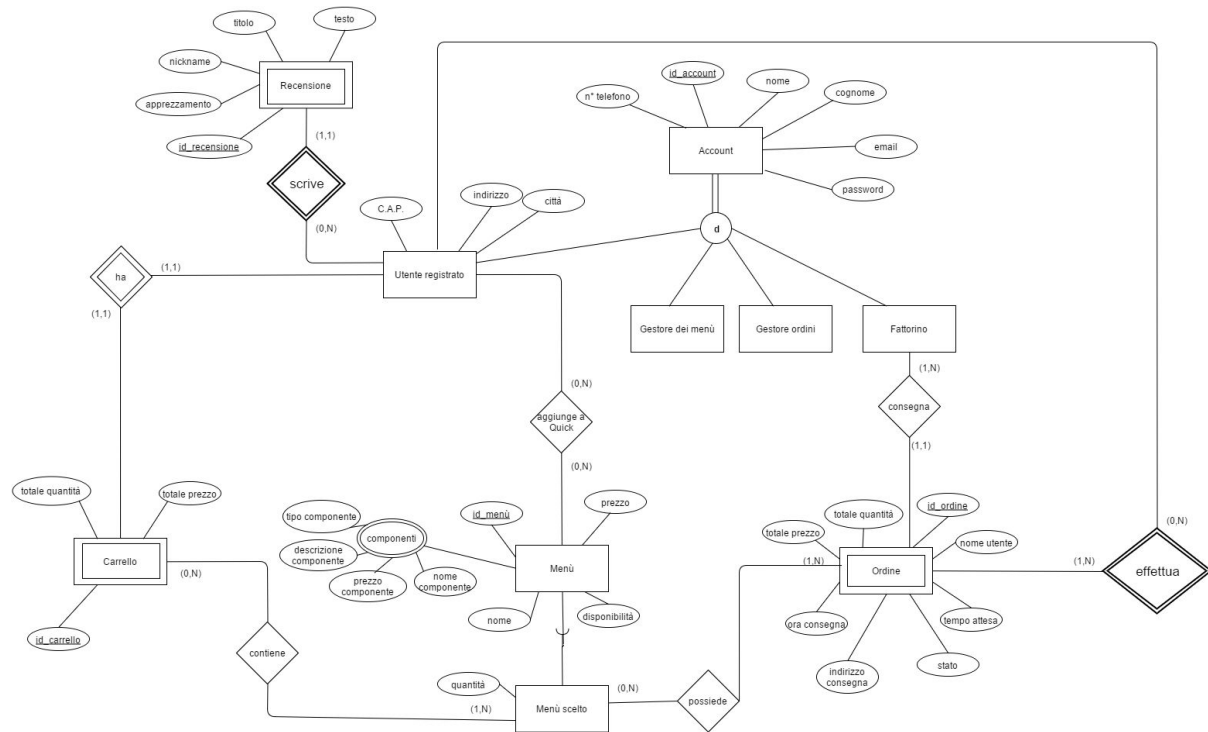
I dati persistenti sono sempre lì e vengono cancellati solo se lo si richiede espressamente. Nel caso specifico, in previsione del gran numero di accessi al database e alla necessità di gestire anche più queries contemporaneamente, si è optati per un DBMS relazionale, MySQL.

La sigla SQL presente all'interno del nome sta a indicare che si tratta di un DBMS in grado di interpretare le istruzioni del linguaggio SQL (utile per le query). MySQL permette la creazione di “database relazionali” ossia consente la conservazione dei dati in tabelle separate anziché in un'unica grande entità. Questa sua particolare caratteristica consente di raggiungere una buona flessibilità e velocità di accesso ai dati ed una maggior modellazione delle basi dati. Semplicità d'uso, robustezza e velocità di esecuzione sono le caratteristiche principali di questo DBMS. La sua struttura multi-thread consta di un thread fisso che controlla le connessioni in ingresso e un thread attivo per ogni connessione. I vari client non devono aspettare che le queries di altri siano soddisfatte, possono lavorare simultaneamente. Il thread che controlla le connessioni impedisce che due thread scrivano sulla stessa tabella nello stesso momento. In ogni tabella all'interno del nostro database sarà possibile accedere ai dati per compiere operazioni quali inserimento, modifica, cancellazione o semplicemente consultazione.

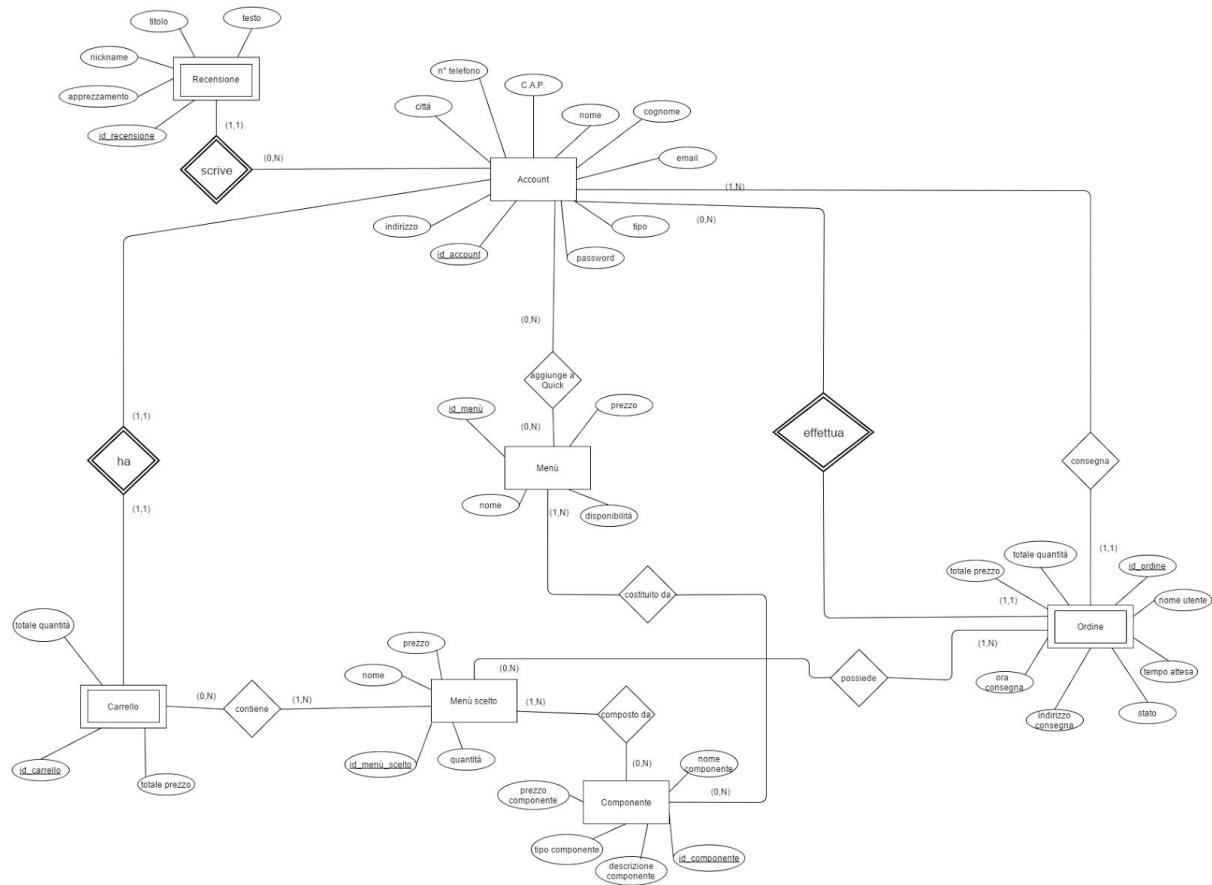
2. Progettazione del database

2.1 Gestione dei dati persistenti

2.1.1 Modello ER



2.1.2 Modello ER ristrutturato



2.1.3 Dizionario dei dati

2.1.3.1 Entità

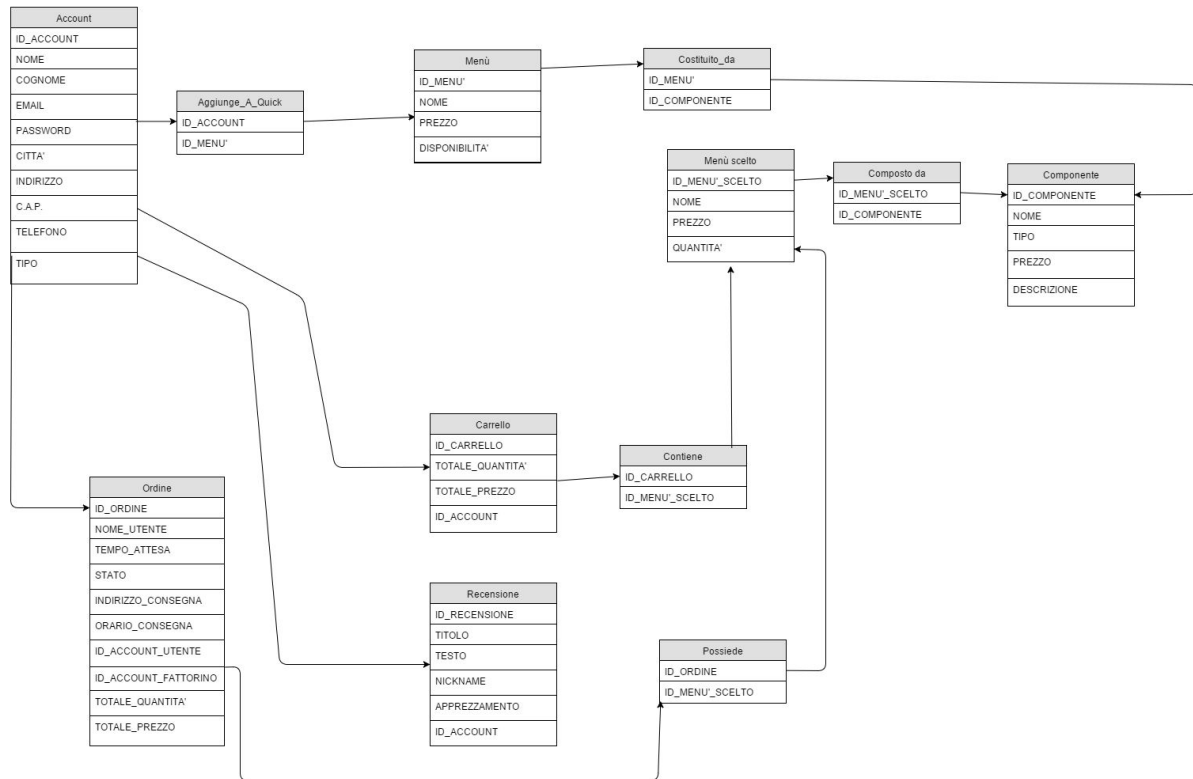
Entità	Descrizione	Attributi	Identificatore
Account	Contiene le informazioni di un account del sistema.	id_account, nome, cognome, email, password, città, indirizzo, CAP, numero di telefono, tipo.	id_account
Menù	Contiene le informazioni di	id_menus, nome, prezzo,	id_menus

	un menù.	disponibilità.	
Menù scelto	Contiene le informazioni di un menù.	id_menù_scelto, nome, prezzo, quantità.	id_menù_scelto
Ordine	Contiene le informazioni di un ordine.	id_ordine, nome utente, tempo attesa, stato, indirizzo consegna, orario consegna, totale quantità, totale prezzo, id_account_utente, id_account_fattorino.	id_ordine, id_account_utente
Carrello	Contiene le informazioni di un carrello.	id_carrello, totale quantità, totale prezzo, id_account.	id_carrello, id_account
Recensione	Contiene le informazioni di una recensione.	id_recensione, titolo, nickname, testo, apprezzamento, id_account	id_recensione, id_account
Componente	Contiene le informazioni di un componente di un menù scelto.	id_componente, nome, prezzo, tipo, descrizione.	id_componente

2.1.3.2 Relazioni

Relazione	Descrizione	Entità coinvolte
Aggiunge a Quick	Descrive la relazione tra un account di un utente registrato e un menù.	Account utente, menù.
Scrive	Descrive la relazione tra un account di un utente registrato e una recensione.	Account utente, recensione.
Ha	Descrive la relazione tra un account di un utente registrato e il carrello.	Account utente, carrello.
Effettua	Descrive la relazione tra un account di un utente registrato e un ordine.	Account utente, ordine.
Contiene	Descrive la relazione tra il carrello e un menù scelto.	Carrello, menù scelto.
Consegna	Descrive la relazione tra un account di un fattorino e un ordine.	Account fattorino, ordine.
Costituito da	Descrive la relazione tra menù e componente	Menù, componente
Composto da	Descrive la relazione tra menù scelto e componente	Menù scelto, componente
Possiede	Descrive la relazione tra ordine e menù scelto	Ordine, menù scelto

2.1.4 Modello logico



3. Descrizione del Database

Tabella: Account

Attributo	Tipo	Chiave	Opzionale
ID_ACCOUNT	INT(10)	PRIMARIA	NO
NOME	VARCHAR(20)		NO
COGNOME	VARCHAR(20)		NO
EMAIL	VARCHAR(25)		NO
PASSWORD	VARCHAR(15)		NO
CITTA'	VARCHAR(25)		SI
INDIRIZZO	VARCHAR(30)		SI

C.A.P.	INT(5)		SI
N° TELEFONO	VARCHAR(10)		NO
TIPO	VARCHAR(20)		NO

Tabella: Aggiunge_A_Quick

Attributo	Tipo	Chiave	Opzionale
ID_ACCOUNT	INT(10)	ESTERNA	NO
ID_MENU'	INT(10)	ESTERNA	NO

Tabella: Menù

Attributo	Tipo	Chiave	Opzionale
ID_MENU'	INT(10)	PRIMARIA	NO
NOME	VARCHAR(20)		NO
PREZZO	FLOAT		NO
DISPONIBILITA'	INT(7)		NO

Tabella: Menù scelto

Attributo	Tipo	Chiave	Opzionale
ID_MENU'_SCELTO	INT(10)	PRIMARIA	NO
NOME	VARCHAR(20)		NO

PREZZO	FLOAT(7)		NO
QUANTITA'	INT(2)		NO

Tabella: Componente

Attributo	Tipo	Chiave	Opzionale
ID_COMPONENTE	INT(10)	PRIMARIA	NO
NOME	VARCHAR(30)		NO
TIPO	VARCHAR(15)		NO
DESCRIZIONE	VARCHAR(250)		SI
PREZZO	FLOAT(7)		SI

Tabella: Composto da

Attributo	Tipo	Chiave	Opzionale
ID_MENU'_SCELTO	INT(10)	ESTERNA	NO
ID_COMPONENTE	INT(10)	ESTERNA	NO

Tabella: Costituito da

Attributo	Tipo	Chiave	Opzionale
ID_MENU'	INT(10)	ESTERNA	NO
ID_COMPONEN	INT(10)	ESTERNA	NO

TE			
----	--	--	--

Tabella: Carrello

Attributo	Tipo	Chiave	Opzionale
ID_CARRELLO	INT(10)	PRIMARIA	NO
TOTALE_QUANTITA'	INT(7)		NO
TOTALE_PREZZO	FLOAT(7)		NO
ID_ACCOUNT	INT(10)	ESTERNA	NO

Tabella : Contiene

Attributo	Tipo	Chiave	Opzionale
ID_CARRELLO	INT(10)	ESTERNA	NO
ID_MENU'_SCELTO	INT(10)	ESTERNA	NO

Tabella : Possiede

Attributo	Tipo	Chiave	Opzionale
ID_ORDINE	INT(10)	ESTERNA	NO
ID_MENU'_SCELTO	INT(10)	ESTERNA	NO

Tabella: Ordine

Attributo	Tipo	Chiave	Opzionale
ID_ORDINE	INT(10)	PRIMARIA	NO
NOME_UTENTE	VARCHAR(20)		NO
TEMPO_ATTESA	VARCHAR(5)		NO
STATO	VARCHAR(20)		SI
INDIRIZZO_CONSEGNA	VARCHAR(30)		NO
ORARIO_CONSEGNA	VARCHAR(5)		NO
TOTALE QUANTITA'	INT(7)		NO
TOTALE PREZZO	FLOAT(7)		NO
ID_ACCOUNT_UTENTE	INT(10)	ESTERNA	NO
ID_ACCOUNT_FATTORINO	INT(10)	ESTERNA	

Tabella: Recensione

Attributo	Tipo	Chiave	Opzionale
ID_RECENSIONE	INT(10)	PRIMARIA	NO
TITOLO	VARCHAR(30)		NO
TESTO	VARCHAR(300)		NO
NICKNAME	VARCHAR(15)		NO
APPREZZAMENTO	FLOAT		NO
ID_ACCOUNT	INT(10)	ESTERNA	NO

4. Codice di creazione del Database

Tabella: Account

```
CREATE TABLE Account(  
    ID_ACCOUNT          int(10)      NOT NULL  
    AUTO_INCREMENT,  
    NOME                varchar(20)   NOT NULL,  
    COGNOME             varchar(20)   NOT NULL,  
    EMAIL               varchar(25)   NOT NULL,  
    PASS                varchar(15)   NOT NULL,  
    CITTÁ               varchar(25)   ,  
    INDIRIZZO           varchar(30)   ,  
    CAP                 int(5)        ,  
    N_TELEFONO          varchar(10)   NOT NULL,  
    TIPO                varchar(20)   NOT NULL,  
    UNIQUE(EMAIL),  
    PRIMARY KEY(ID_ACCOUNT)  
);
```

Tabella: Recensione

```
CREATE TABLE Recensione(  
    ID_RECENSIONE       int(10)      NOT NULL  
    AUTO_INCREMENT,  
    TITOLO              varchar(30)   NOT NULL,  
    TESTO               varchar(300)  NOT NULL,  
    NICKNAME            varchar(15)   NOT NULL,  
    APPREZZAMENTO       float        NOT NULL,  
    ID_ACCOUNT          int(10)      NOT NULL,  
    PRIMARY KEY(ID_RECENSIONE, ID_ACCOUNT),  
    FOREIGN KEY(ID_ACCOUNT) references  
    Account(ID_ACCOUNT)  
    ON DELETE CASCADE ON UPDATE CASCADE
```

);

Tabella: Aggiunge_A_Quick

```
CREATE TABLE Aggiunge_A_Quick(  
    ID_ACCOUNT      int(10)          NOT NULL,  
    ID_MENÙ         int(10)          NOT NULL,  
    PRIMARY KEY(ID_ACCOUNT, ID_MENÙ),  
    FOREIGN KEY(ID_ACCOUNT) references  
    Account(ID_ACCOUNT),  
    FOREIGN KEY(ID_MENÙ) references Menù(ID_MENÙ)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Tabella: Ordine

```
CREATE TABLE Ordine(  
    ID_ORDINE                int(10)          NOT  
    NULL  
    AUTO_INCREMENT,  
    NOME_UTENTE              varchar(20)      NOT NULL,  
    TEMPO_ATTESA             varchar(5)       NOT NULL,  
    STATO                    varchar(20)      NOT NULL,  
    INDIRIZZO_CONSEGNA       varchar(30)      NOT  
    NULL,  
    ORARIO_CONSEGNA          varchar(5)       NOT NULL,  
    TOTALE_QUANTITÀ          int(7)          NOT NULL default 0,  
    TOTALE_PREZZO            float(7)        NOT NULL default 0,  
    ID_ACCOUNT_UTENTE        int(10)          NOT NULL,  
    ID_ACCOUNT_FATTORINO     int(10)          ,  
    PRIMARY KEY(ID_ORDINE, ID_ACCOUNT_UTENTE),  
    FOREIGN KEY(ID_ACCOUNT_UTENTE) references  
    Account(ID_ACCOUNT)  
    ON DELETE CASCADE ON UPDATE CASCADE,
```

FOREIGN KEY(ID_ACCOUNT_FATTORINO) references
Account(ID_ACCOUNT)
ON DELETE CASCADE ON UPDATE CASCADE

);

Tabella: Carrello

```
CREATE TABLE Carrello(  
    ID_CARRELLO          int(10)          NOT NULL  
    AUTO_INCREMENT,  
    ID_ACCOUNT           int(10)          NOT NULL,  
    TOTALE_QUANTITÀ      int(7)           NOT NULL,  
    TOTALE_PREZZO        float(7)         NOT NULL,  
    PRIMARY KEY(ID_CARRELLO, ID_ACCOUNT),  
    FOREIGN KEY(ID_ACCOUNT) references  
    Account(ID_ACCOUNT)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Tabella: Menù

```
CREATE TABLE Menù(  
    ID_MENÙ              int(10)          NOT NULL  
    AUTO_INCREMENT,  
    NOME                  varchar(20)      NOT NULL,  
    PREZZO                float            NOT NULL,  
    DISPONIBILITÀ        int(7)           NOT NULL,  
    UNIQUE(MENÙ'),  
    PRIMARY KEY(ID_MENÙ)  
);
```

Tabella: Menù scelto


```

CREATE TABLE Menù_scelto(
    ID_MENÙ_SCELTO      int(10)          NOT NULL
    AUTO_INCREMENT,
    NOME                 varchar(20)       NOT NULL,
    PREZZO               float(7)         NOT NULL,
    QUANTITÀ             int(2)           NOT NULL,
    PRIMARY KEY(ID_MENÙ_SCELTO)
);

```

Tabella: Componente

```

CREATE TABLE Componente(
    ID_COMPONENTE        int(10)          NOT NULL
    AUTO_INCREMENT,
    NOME                 varchar(30)       NOT NULL,
    TIPO                 varchar(15)       NOT NULL,
    DESCRIZIONE          varchar(250),
    PREZZO               float(7),
    UNIQUE(NOME),
    PRIMARY KEY(ID_COMPONENTE)
);

```

Tabella: Contiene

```

CREATE TABLE Contiene(
    ID_CARRELLO          int(10)          NOT NULL,
    ID_MENÙ_SCELTO       int(10)          NOT NULL,
    PRIMARY KEY (ID_CARRELLO, ID_MENÙ_SCELTO),
    FOREIGN KEY(ID_CARRELLO ) references
    Carrello(ID_CARRELLO)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(ID_MENÙ_SCELTO) references Menù
    scelto(ID_MENÙ_SCELTO)
);

```

ON DELETE CASCADE ON UPDATE CASCADE

);

Tabella: Composto da

```
CREATE TABLE Composto_da(  
    ID_MENU'_SCELTO    int(10)          NOT NULL,  
    ID_COMPONENTENTE    int(10)          NOT NULL,  
    PRIMARY KEY (ID_MENU'_SCELTO, ID_COMPONENTENTE),  
    FOREIGN KEY(ID_MENU'_SCELTO) references Menu'  
scelto(ID_MENU'_SCELTO) ON DELETE CASCADE ON UPDATE  
CASCADE,  
    FOREIGN KEY(ID_COMPONENTENTE ) references  
Componente(ID_COMPONENTENTE)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Tabella: Costituito da

```
CREATE TABLE Costituito_da(  
    ID_MENU'          int(10)          NOT NULL,  
    ID_COMPONENTENTE    int(10)          NOT NULL,  
    PRIMARY KEY (ID_MENU', ID_COMPONENTENTE),  
    FOREIGN KEY(ID_MENU') references Menu' (ID_MENU') ON  
DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(ID_COMPONENTENTE ) references  
Componente(ID_COMPONENTENTE)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Tabella: Possiede

```
CREATE TABLE Possiede (  
    ID_ORDINE          int(10)          NOT NULL,  
    ID_MENU'_SCELTO    int(10)          NOT NULL,
```

```
PRIMARY KEY (ID_ORDINE, ID_MENU'_SCELTO),  
FOREIGN KEY(ID_MENU'_SCELTO ) references  
Menù_scelto(ID_MENÙ_SCELTO) ON DELETE CASCADE ON  
UPDATE CASCADE,  
FOREIGN KEY(ID_ORDINE) references Ordine(ID_ORDINE)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

5. DESIGN DELLE TRANSAZIONI

5.1 GESTIONE ACCOUNT

Registrazione di un utente

```
INSERT INTO Account(NOME, COGNOME, EMAIL, PASS,  
CITTÁ, INDIRIZZO, CAP, N_TELEFONO, TIPO)  
VALUES ('EGIDIO', 'GIACOA', 'E.GIACOA@HOTMAIL.IT',  
'EGIDIOGIACOA', 'SALERNO', 'VIA ROMA 15', 85040,  
3284874895, 'UTENTE');
```

Modifica dei dati di un utente

```
update account  
set pass='EGIDIO1994', indirizzo='VIA MARONE  
38', n_telefono=3891043119  
where id_account=1;
```

5.2 GESTIONE MENU'

Inserimento di un menù

```
INSERT INTO Menù(NOME, PREZZO, DISPONIBILITA)  
VALUES ('CLASSICO', 8.00, 50);
```

Visualizzazione della lista dei menù

```
select *
```

from menù;

Aggiunta di un menù a quick menù

```
INSERT INTO Aggiunge_a_quick(ID_ACCOUNT, ID_MENÙ)  
VALUES (1,2);
```

Acquisto del menù

```
INSERT INTO MENÙ_SCELTO(NOME,PREZZO,QUANTITÀ)  
VALUES ('CLASSICO',8.00,2);
```

```
INSERT INTO Contiene(ID_CARRELLO, ID_MENÙ_SCELTO)  
VALUES (1,1);
```

Rimozione di un menù

```
delete  
from menù  
where id_menù=3;
```

Modifica di un menù

```
delete  
from costituito_da  
where id_menù=5 and id_componente=10;
```

```
INSERT INTO Costituito_da(ID_MENÙ, ID_COMPONENTE)  
VALUES(1,21);
```

```
update menù  
set disponibilità=30, prezzo=10  
where id_menù=1;
```

5.3 GESTIONE CARRELLO

Visualizzazione del carrello

```
select totale_quantità, totale_prezzo, id_menù_scelto  
from carrello,contiene;
```

Rimozione di un menù dal carrello

```
delete from contiene  
where id_menù_scelto=1 and id_carrello=1;
```

Svuotamento del carrello

```
delete  
from contiene  
where id_carrello=1;
```

Completamento di un ordine

```
INSERT INTO Ordine ( NOME_UTENTE, TEMPO_ATTESA,  
STATO, INDIRIZZO_CONSEGNA, ORARIO_CONSEGNA,  
ID_ACCOUNT_UTENTE) VALUES ( 'EGIDIO', '21:30', 'ATTESA',  
'VIA ROMA 15', '21:00', 1);
```

5.4 GESTIONE ORDINI

Inserimento di un ordine in stato “In preparazione”

```
INSERT INTO Ordine (NOME_UTENTE, TEMPO_ATTESA,  
STATO, INDIRIZZO_CONSEGNA, ORARIO_CONSEGNA,  
ID_ACCOUNT_UTENTE, ID_ACCOUNT_FATTORINO) VALUES (  
'EGIDIO', '21:30', 'IN PREPARAZIONE', 'VIA ROMA 15', '21:00',  
1, 1);
```

Modifica dello stato dell'ordine in "Consegna"

```
update ordine  
set stato='consegna'  
where id_account_utente=1 and id_ordine=1;
```

Inserimento di un ordine in stato "Ritardo"

```
INSERT INTO Ordine (NOME_UTENTE, TEMPO_ATTESA,  
STATO, INDIRIZZO_CONSEGNA, ORARIO_CONSEGNA,  
ID_ACCOUNT_UTENTE, ID_ACCOUNT_FATTORINO) VALUES  
(EGIDIO, '21:30', 'RITARDO', 'VIA ROMA 15', '21:00', 1, 1);
```

Modifica di uno stato dell'ordine in "Evaso"

```
update ordine  
set stato='evaso'  
where id_account_fattorino=1 and id_ordine=1 and  
id_account_utente=1;
```

Visualizzazione degli ordini

```
select *  
from ordine;
```

Modifica tempo di attesa nella coda

```
update ordine  
set tempo_attesa='22:00'  
where id_account_utente=1 and id_ordine=1;
```

Cancellazione di un ordine

```
delete from ordine  
where id_account_utente=1 and id_ordine= 1;
```

5.5 GESTIONE RECENSIONI

Visualizzazione delle recensioni

```
select *  
from recensione;
```

Aggiunta di una recensione

```
INSERT INTO Recensione( TITOLO, TESTO,  
NICKNAME, APPREZZAMENTO, ID_UTENTE) VALUES  
( 'OTTIMI PANINI', 'HO PRESO DEI PANINI  
ECCEZIONALI, CONSEGNA VELOCE, PROVARE PER  
CREDERE', 'EGIDIO', 5, 1 );
```

5.6 GESTIONE QUICK MENU'

Visualizzazione del quick menù

```
select *  
from aggiunge_a_quick;
```

Rimozione di un menù dal quick menù

```
delete from aggiunge_a_quick  
where id_menù= 4;
```