

## **Ristorante LaBaia**

Renato Esposito	0124001881
Vittorio Piscitelli	0124002000
Mario Oliva	0124001906

**Data di consegna**

22/07/2020

**Anno accademico**

2019 - 2020

# INDICE

## Progettazione

Sintesi dei requisiti .....	6
Diagramma EE/R .....	7
Diagramma Relazionale .....	8
Utenti e le loro categorie .....	8
Operazioni degli utenti .....	9
Volumi .....	12
Vincoli di integrità .....	13
Statici .....	13
Dinamici .....	13
Verifica di normalità .....	13
Possibili estensioni .....	14

## Implementazione

Creazioni utenti .....	14
Data Definition Language .....	14
Cliente .....	14
Ristorante .....	15
Prenotazione .....	15
Staff .....	16
Lavora .....	16
Journey .....	16
Ordine .....	17
Ordine casa .....	17
Tavolo .....	17
Ordine sala .....	17
Consegna .....	18
Carta di credito .....	18
Conto .....	18
Menu .....	19
Seleziona .....	19

Pasto .....	19
Contiene .....	19
Ingrediente .....	19
Si compone .....	20
Fornitore .....	20
Fornitura .....	20
Proviene da .....	21
Data Manipulation Language .....	21
Trigger .....	47
No same date .....	47
Un ristorante massimo .....	48
Massimo sedici dipendenti .....	48
Controllo tavolo ristorante .....	49
Controllo prenotazione .....	49
Controllo journey .....	50
Controllo giorno libero .....	51
Controllo data carta di crediti .....	51
Controllo capienza tavolo .....	52
Procedure e funzioni .....	52
Seleziona menu .....	52
Pren .....	53
Disdetta .....	55
Assegna tavolo .....	55
Get conto .....	57
Get fatturato .....	59
Viste .....	59
Ordini da effettuare .....	59
Prenotazioni totali .....	60
Data Control Language .....	60

## Elenco delle figure

1	Diagramma Entità-Associazione .....	7
---	-------------------------------------	---

2	Diagramma relazionale .....	8
3	Diagramma UML dei casi d'uso .....	9

## Elenco delle tabelle

1	Tavola degli utenti .....	8
2	Tavola delle operazioni .....	9
3	Tavola dei volumi .....	12

## Elenco degli script

CreazioniUtenti.sql .....	14
ClienteDDL.sql .....	14
RistoranteDDL.sql .....	15
PrenotazioneDDL.sql .....	15
StaffDDL.sql .....	16
LavoraDDL.sql .....	16
JourneyDDL.sql .....	16
OrdineDDL.sql .....	17
OrdineCasaDDL.sql .....	17
TavoloDDL.sql .....	17
OrdineSalaDDL.sql .....	17
ConsegnaDDL.sql .....	18
CartaDiCreditoDDL.sql .....	18
ContoDDL.sql .....	18
MenuDDL.sql .....	19
SelezionaDDL.sql .....	19
PastoDDL.sql .....	19
ContieneDDL.sql .....	19
IngredienteDDL.sql .....	19
SiComponeDDL.sql .....	20
FornitoreDDL.sql .....	20
FornituraDDL.sql .....	20
ProvieneDaDDL.sql .....	21

ClienteDML.sql .....	21
RistoranteDML.sql .....	21
PrenotazioniDML.sql .....	21
StaffDML.sql .....	24
LavoraDML.sql .....	26
JourneyDML.sql .....	28
OrdineDML.sql .....	29
TavoloDML.sql .....	30
OrdineSalaDML.sql .....	31
OrdineCasaDML.sql .....	32
ConsegnaDML.sql .....	32
CartaDiCreditoDML.sql .....	33
ContoDML.sql .....	34
MenuDML.sql .....	35
SelezionaDML.sql .....	35
PastoDML.sql .....	37
ContieneDML.sql .....	37
IngredienteDML.sql .....	38
SiComponeDML.sql .....	39
FornitoreDML.sql .....	41
FornituraDML.sql .....	42
ProvieneDaDML.sql .....	42
NoSameDateTrigger.sql .....	47
UnRistoranteMassimoTrigger.sql .....	48
MassimoSediciDipendentiTrigger.sql .....	48
ControlloTavoloRistoranteTrigger.sql .....	49
ControlloPrenotazioneTrigger.sql .....	49
ControlloJourneyTrigger.sql .....	50
ControlloGiornoLiberoTrigger.sql .....	51
ControlloDataCartaDiCreditoTrigger.sql .....	51
ControlloCapienzaTavoloTrigger.sql .....	52
SelezionaMenuProc.sql .....	52
PrenProc.sql .....	53
DisdettaProc.sql .....	55

AssegnaTavoloProc.sql .....	55
GetContoProc.sql .....	57
GetFatturatoProc.sql .....	59
OrdiniDaEffettuareView.sql .....	59
PrenotazioniTotaliView.sql .....	60
CustomerDCL.sql .....	60
FattorinoDCL.sql .....	60
CameriereDCL.sql .....	60
CassiereDCL.sql .....	61
ChefDCL.sql .....	61
ProprietarioDCL.sql .....	61

## Progettazione

Il proprietario della catena di ristoranti “LaBaia” vuole automatizzare la gestione del sistema di prenotazione nonché dell’intera organizzazione della catena. In seguito ad una discussione in merito agli obiettivi da perseguire, sono emersi i seguenti requisiti.

### Sintesi dei requisiti

La base di dati deve gestire le prenotazioni dei clienti e memorizzare alcuni aspetti amministrativi di ogni ristorante appartenente alla catena, quali: organizzazione dello staff, tracciabilità della consegna, informazioni relative ai pasti e alla loro provenienza. I clienti, prima di effettuare una prenotazione, devono, se necessario, registrarsi inserendo i propri dati anagrafici e successivamente riferire, al ristorante scelto, con non più di due settimane di anticipo, la modalità di consumazione, cioè se consumare in sede o ricevere il pasto a domicilio, infine possono disdire tale prenotazione solo se quest’ultima prevede la consumazione in sede. Ciascun ristorante ha un unico giorno di chiusura, non nel week-end e tavoli con capienza non superiore alle quindici persone. Nel caso in cui si tratta di una consumazione a domicilio, al cliente viene assegnato e comunicato un numero d’ordine con il quale può selezionare uno o più menù, quest’ultimi sono prefissati e quindi non c’è possibilità di modificare i pasti al loro interno, infine un fattorino si occupa di consegnare l’ordine e memorizzare le informazioni relative al viaggio. Nel caso in cui si tratta di una consumazione in sala e il cliente si presenta effettivamente nella data stabilita, non ritardando per più di trenta minuti, un cameriere assegnerà un tavolo e un numero d’ordine, successivamente il cliente potrà selezionare uno o più menù. In entrambi i casi di consumazione il pagamento deve essere effettuato rigorosamente con carta di credito che per motivi fiscali deve essere memorizzata all’interno della base di dati. Infine, è richiesta anche la memorizzazione dello scarico merci da parte di un fornitore, che può essere di volta in volta diverso, al fine di poter tener conto del numero di ingredienti in deposito per preparare i vari pasti dei menu.

## Diagrammi EE/R

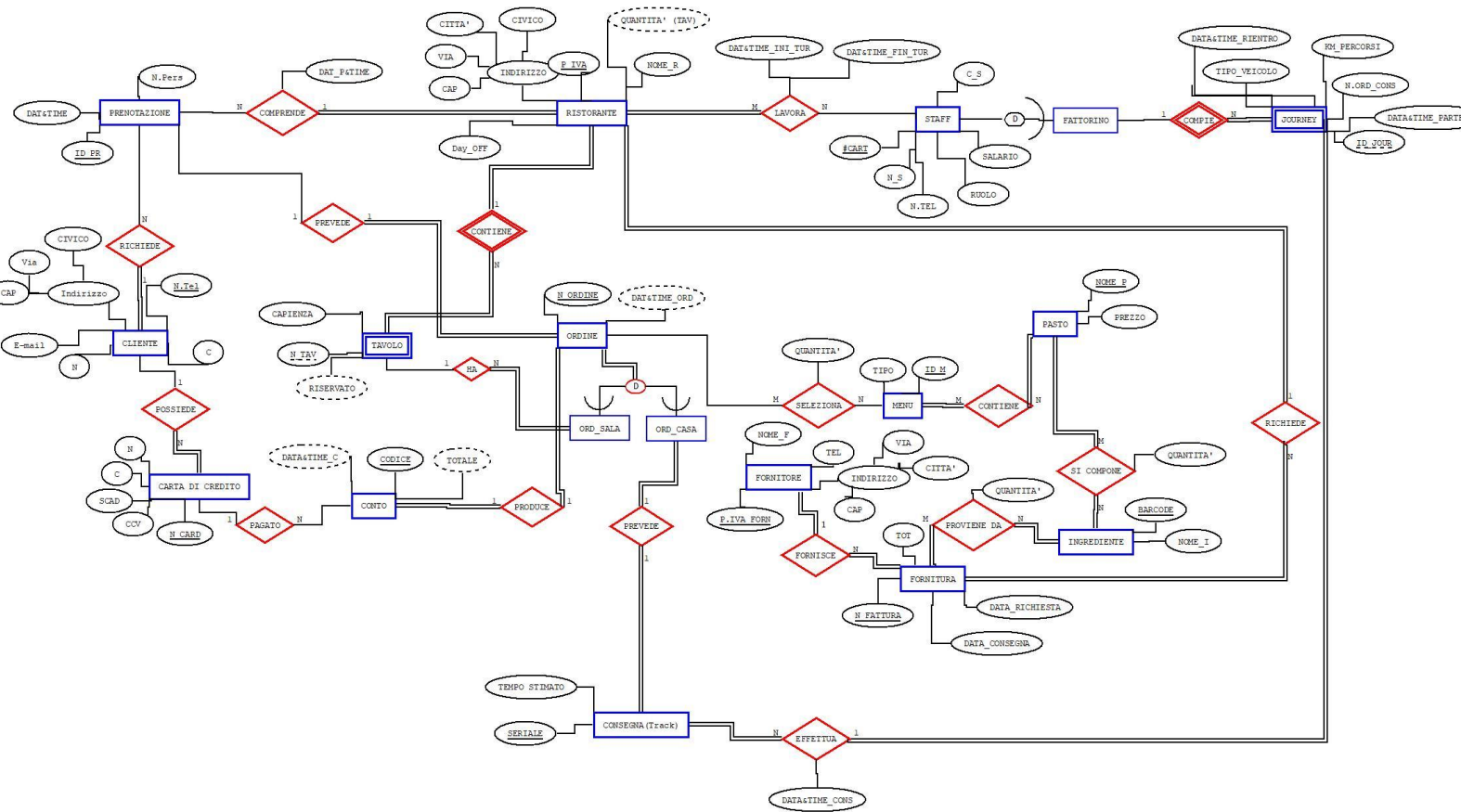


Figura 1: Diagramma Entità-Associazione

## Diagramma Relazionale

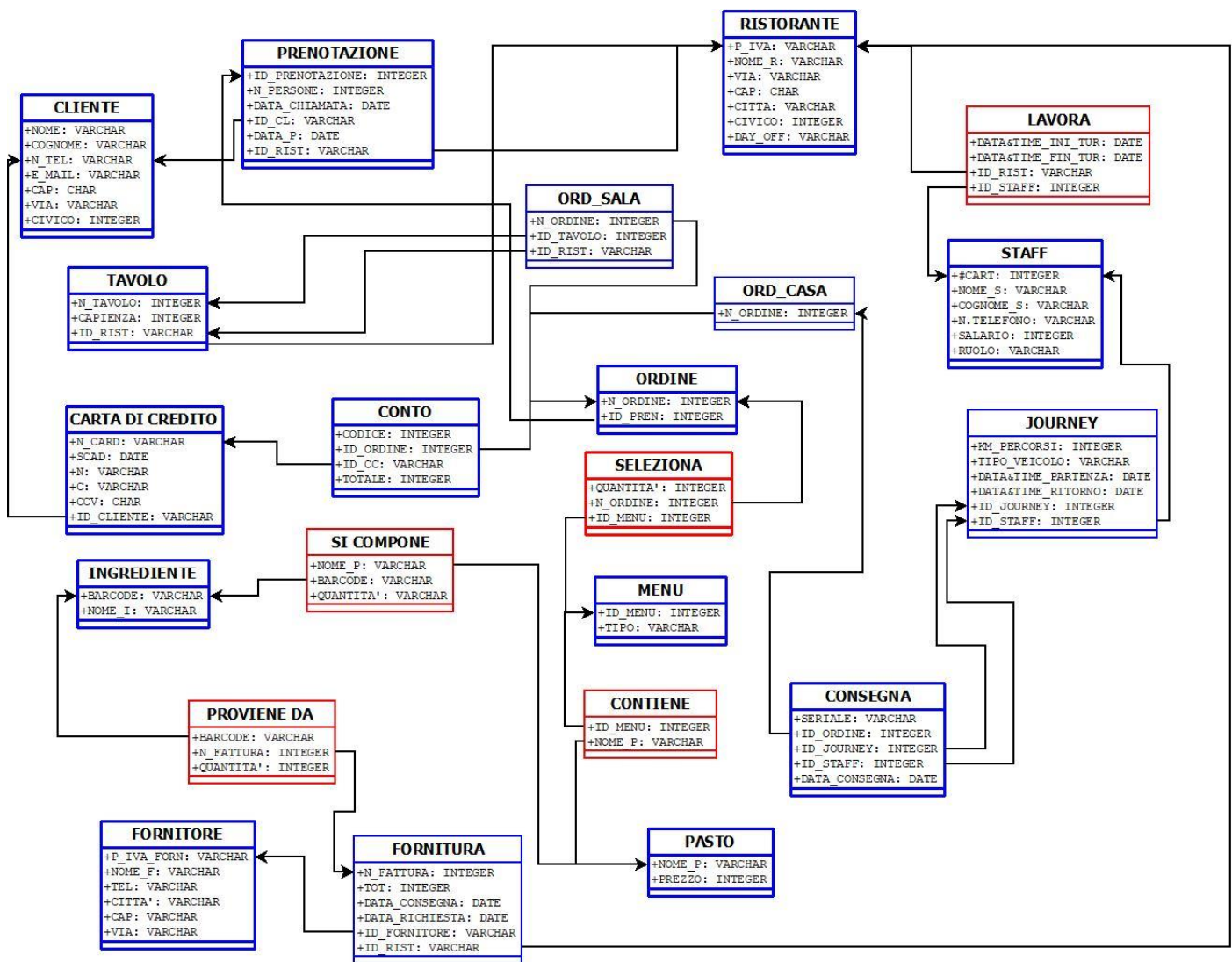


Figura 2: Diagramma relazionale

## Utenti e le loro categorie

All'interno della nostra base di dati prevediamo la presenza di 5 utenti, i quali con permessi differenti potranno accedere a informazioni differenti.

Utente	Tipo	Volume	Permessi
Progetto	Amministratore	1	ALL
Customer	Comune	1	EXECUTE ON PREN EXECUTE ON DISDETTA EXECUTE ON SELEZIONA_MENU INSERT ON CLIENTE
Fattorino	Comune	1	SELECT ON INFO_PER_CONSEGNA
Cameriere	Comune	1	EXECUTE ON ASSEGNA_TAVOLO ALL ON SELEZIONA SELECT ON PRENOTAZIONE EXECUTE ON SELEZIONA_MENU



Proprietario	Comune	1	<a href="#">EXECUTE ON GET_FATTURATO</a> <a href="#">SELECT ON INFO_PREN</a>
Cassiere	Comune	1	<a href="#">EXECUTE ON GET_CONTO</a>

Tabella 1: Tavola degli utenti

## Operazioni Utenti

### Diagramma UML dei casi d'uso

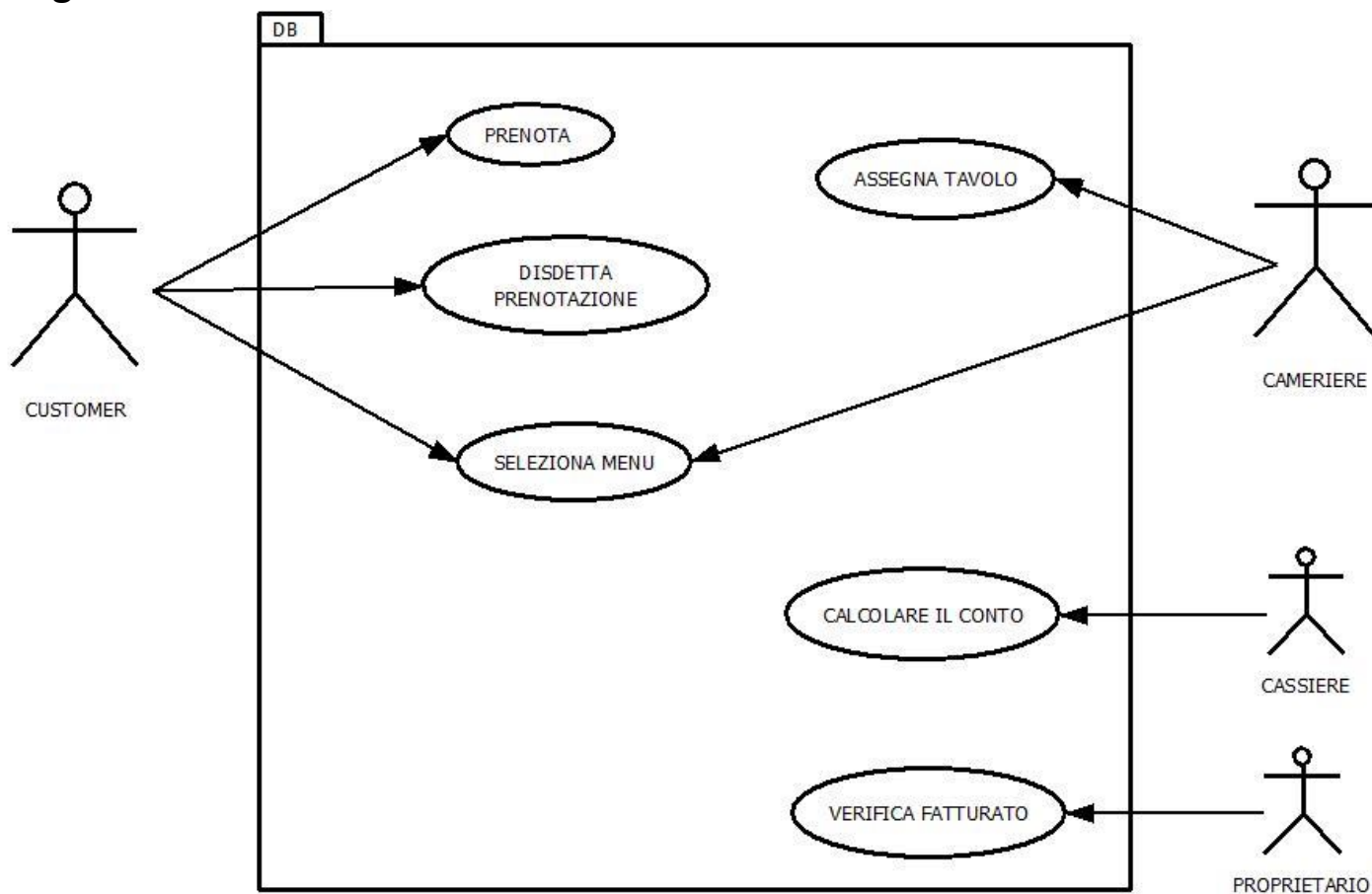


Figura 3: Diagramma UML dei casi d'uso

Schede descrittore operazione per le operazioni degli utenti:

<i>Operazione</i>	<b>Pren()</b>
<i>Scopo</i>	Automatizzare la prenotazione di un utente customer
<i>Argomenti</i>	Pk cliente, numero persone della prenotazione, data e ora prenotazione, id del ristorante per cui si vuole prenotare e un flag (0,1) per indicare se si vuole prenotare un pasto d asporto o un tavolo in sala.
<i>Risultato</i>	Aggiunta della prenotazione o messaggio d'errore
<i>Errori</i>	Flag errato, ristorante errato
<i>Usa</i>	Cliente, ristorante, prenotazione , ordine e ordine_casa
<i>Modifica</i>	Prenotazione, ordine e ordine_sala
<i>Prima</i>	L'utente non risulta prenotato
<i>Poi</i>	All'utente viene assegnato un n. ordine, per prenotare un pasto da casa e il codice univoco della prenotazione

<i>Operazione</i>	<b>Assegna_tavolo()</b>
<i>Scopo</i>	Assegnare un tavolo ad un utente cameriere che ha effettuato la prenotazione
<i>Argomenti</i>	Pk cliente e id prenotazione
<i>Risultato</i>	Assegnazione di un tavolo e di un n. ordine per poter selezionare i vari menu
<i>Errori</i>	Troppo tardi, troppo presto
<i>Usa</i>	Prenotazione, tavolo, ordine e ordine_sala
<i>Modifica</i>	Tavolo, ordine e ordine_sala
<i>Prima</i>	L'utente non risulta avere ne un tavolo ne un n. ordine
<i>Poi</i>	All'utente viene assegnato un n. ordine, per prenotare un pasto da casa e il tavolo.

<i>Operazione</i>	<b>Seleziona_menu()</b>
<i>Scopo</i>	Permette all'utente customer o cameriere di selezionare uno o piu menu
<i>Argomenti</i>	Pk cliente, id del menu e la quantita di quel menu
<i>Risultato</i>	Assegnazione di uno o piu menu ad ogni utente di un determinato tavolo
<i>Errori</i>	Errore_1(Utente non prenotato)
<i>Usa</i>	Prenotazione, ordine e seleziona

<i>Modifica</i>	Seleziona
<i>Prima</i>	All'ordine dell'utente non risultano essere assegnati menu
<i>Poi</i>	All'ordine dell'utente risultano uno o piu menu assegnati

<i>Operazione</i>	<b>Disdetta()</b>
<i>Scopo</i>	Permette all'utente customer di disdire la prenotazione (a determinate condizioni)
<i>Argomenti</i>	Pk cliente e data della prenotazione
<i>Risultato</i>	Cancellazione di una prenotazione
<i>Errori</i>	Ordine_da_casa
<i>Usa</i>	Prenotazione, ordine_casa e ordine
<i>Modifica</i>	Prenotazione
<i>Prima</i>	Era presente una determinata prenotazione
<i>Poi</i>	La stessa prenotazione non è piu presente

<i>Operazione</i>	<b>Get_conto()</b>
<i>Scopo</i>	Permette all'utente cassiere di generare il conto da far pagare all'utente customer
<i>Argomenti</i>	N. ordine
<i>Risultato</i>	Output del conto e inserimento nella tabella
<i>Errori</i>	Conto_inesistente
<i>Usa</i>	Ordine, seleziona e pasto
<i>Modifica</i>	Conto
<i>Prima</i>	Il conto era settato a NULL
<i>Poi</i>	Il conto ha un valore

<i>Operazione</i>	<b>Get_fatturato()</b>
<i>Scopo</i>	Permette all'utente proprietario di ottenere il fatturato di uno dei suoi ristoranti
<i>Argomenti</i>	Id ristorante
<i>Risultato</i>	Stampa fatturato
<i>Errori</i>	No_data_found
<i>Usa</i>	Conto, prenotazione e ordine
<i>Modifica</i>	-

<i>Prima</i>	-
<i>Poi</i>	Stampa a video il fatturato

## Tavola Delle Operazioni

Prevediamo che per l'intera catena nell'arco di una settimana potrebbero esserci all'incirca seicento prenotazioni, assegna tavolo e calcolo conto, trecento disdette al mese, quattromila seleziona menu e infine 4 calcoli del fatturato al mese.

Operazione	Tipo	Volume	Periodo
Prenotazione	B	600	Settimana
Assegna tavolo	B	600	Settimana
Disdetta prenotazione	B	300	Mese
Seleziona Menu	B	2000	Settimana
Calcolo conto	B	600	Settimana
Calcolo fatturato	B	4	Mese

*Tabella 2: Tavola delle operazioni*

## Volumi

Tabella	Tipo	Volume	Incremento	Periodo
Cliente	E	600	200	Mese
Prenotazione	E	800	300	Mese
Ristorante	E	4	0	Anno
Lavora	A	64	0	Anno
Staff	E	64	0	Anno
Journey	ED	600	400	Mese
Consegna	E	600	400	Mese
Ordine	E	2400	1000	Mese
Tavolo	ED	40	0	Anno
Conto	E	800	300	Mese
Carta di credito	E	600	200	Mese
Menu	E	5	0	Anno
Seleziona	A	8000	4000	Mese
Pasto	E	18	0	Anno
Contiene	A	24	0	Anno
Si compone	A	78	0	Anno
Ingrediente	E	45	0	Anno
Proviene da	A	2000	0	Anno
Fornitore	E	20	0	Anno
Fornitura	E	12	0	Anno

*Tabella 3: Tavola dei volumi*

## Vincoli Di Integrità

### Statici

- Il tavolo associato ad un ordine deve essere capiente almeno quanto il numero di persone in prenotazione
- Un cliente non può prenotare nel giorno di chiusura del ristorante
- Il tavolo associato ad una prenotazione deve necessariamente appartenere al ristorante per cui si sta effettuando la prenotazione
- La carta di credito di credito al momento del pagamento non deve essere scaduta
- Il numero di persone in prenotazione deve essere minore o uguale a 15
- I ruoli previsti per il nostro staff sono: cassiere, fattorino, inserviente, lavapiatti, chef e cameriere
- Nessun dipendente può ricevere una paga inferiore a 850 euro
- Il giorno di chiusura del ristorante non può essere compreso tra giovedì e domenica

### Dinamici

- Un cliente non può effettuare più di una prenotazione nella stessa data
- Un dipendente può lavorare al più per un ristorante
- Un ristorante può assumere massimo 16 dipendenti
- Gli ordini effettuati devono essere necessariamente pagati con una cc intestata a nome del cliente che ha fatto l'ordine
- Le consegne devono essere effettuate esclusivamente dai fattorini
- Un cliente può prenotarsi solo per orari compresi tra le 12 e le 23

## Verifica Di Normalità

### Prima forma normale

La prima forma normale è rispettata poiché le tuple all'interno delle tabelle rispettano l'atomicità dei dati e tutte le colonne contengono valori dello stesso tipo.

### Seconda forma normale

La seconda forma normale è rispettata poiché in Journey e tavolo, che hanno chiavi multi-attributo, tutti i loro attributi non chiave sono funzionalmente dipendenti dall'intera chiave composta e non solo da una parte di essa.

### Terza forma normale

La terza forma normale potrebbe non sembrare essere rispettata su carta di credito poiché oltre alla chiave del prenotante, si memorizza anche il nome ed il cognome del titolare della carta, che potrebbe risultare uguale al nome e cognome del prenotante associato alla chiave esterna e quindi si creerebbe ridondanza. Di fatto, in un caso reale, un cliente potrebbe pagare con una carta di credito intestata ad altri per cui abbiamo ritenuto opportuno memorizzare, in ogni caso, i dati dell'intestatario della carta.

Nel resto dello schema la 3NF è rispettata in quanto tutti gli attributi che non sono chiave primaria dipendono esclusivamente dalla chiave stessa.

## Possibili Estensioni

- Tener conto giornalmente delle presenze e dell'assenza dei lavoratori
- Licenziare i lavoratori che superano una certa quantità di assenze annuali
- All'aumentare del numero medio di prenotazioni effettuate in un mese si aumenti il numero di componenti dello staff
- Tener conto di un deposito e di automatizzare la richiesta di fornitura se la quantità di un certo ingrediente è inferiore ad una certa soglia
- Una volta disdetta una prenotazione non è più possibile prenotarsi per lo stesso giorno

# Implementazione

## Creazione utenti

CREATE USER progetto IDENTIFIED BY ristorante;

CREATE USER customer IDENTIFIED BY ristorante;

CREATE USER cameriere IDENTIFIED BY ristorante;

CREATE USER fattorino IDENTIFIED BY ristorante;

CREATE USER proprietario IDENTIFIED BY ristorante;

CREATE USER cassiere IDENTIFIED BY ristorante;

CREATE USER chef IDENTIFIED BY ristorante;

GRANT ALL PRIVILEGES TO progetto;

## Data Definition Language

### Cliente

La tabella cliente è popolata dal cliente stesso che deve registrarsi per poter effettuare la prenotazione, per cui sono necessari i vincoli al fine di inserire correttamente i formati giusti per e-mail e telefono

CREATE TABLE CLIENTE (

NOME VARCHAR (30) NOT NULL,

COGNOME VARCHAR (30) NOT NULL,

N\_TEL VARCHAR (11) PRIMARY KEY,

```
VIA VARCHAR (50) NOT NULL,
CIVICO NUMBER(5,0) NOT NULL,
E_MAIL VARCHAR (60) NOT NULL,
CAP VARCHAR (6) NOT NULL,
CONSTRAINT EMAILCHECKCHECK( REGEXP_LIKE (E_MAIL,'^[A-Za-z]+[A-Za-z0-9._-]+@[A-Za-z0-9.-]' ) ),
CONSTRAINT telefonocheck CHECK( REGEXP_LIKE (N_TEL,'^[0-9]{3}+$') );
```

## Ristorante

Ristorante è popolata dal proprietario e sono necessari i vincoli sul formato della partita iva e sui giorni di chiusura permessi, identificato non come stringhe di caratteri ma come numeri interi che indicano i giorni della settimana.

```
CREATE TABLE RISTORANTE (
P_IVA VARCHAR (20) PRIMARY KEY,
NOME_R VARCHAR (20) NOT NULL,
VIA VARCHAR (50) NOT NULL,
CAP VARCHAR (6) NOT NULL,
CITTA VARCHAR (15) NOT NULL,
CIVICO NUMBER (5,0) NOT NULL,
DAY_OFF CHAR (1) NOT NULL,
CONSTRAINT CHECK_IVA CHECK(REGEXP_LIKE(P_IVA, '^[A-Z]{2}+[A-Z0-9]{9}' ) )
CONSTRAINT CHECK_GIORNO_FESTA CHECK(DAY_OFF >=1 AND DAY_OFF <=3);
);
```

## Prenotazione

La tabella prenotazione è popolata dal cliente attraverso la procedura "pren()", la quale controlla se l'utente non è ancora registrato e se il ristorante selezionato appartiene alla catena. Al fine di evitare errori di inserimento sono necessari i constraint sulla data chiama che deve essere inferiore alla data in cui si vuole effettuare la prenotazione, sul numero di persone che giungeranno al ristorante e infine occorre accertarsi che l'ora in cui si vuole consumare sia tra le 12 e le 23.

```
CREATE TABLE PRENOTAZIONE (
ID_PRENOTAZIONE NUMBER(3,0) PRIMARY KEY,
N_PERSONE NUMBER(2,0) NOT NULL,
ID_CL VARCHAR (11) NOT NULL,
DATA_CHIAMATA DATE,
ID_RISTORANTE VARCHAR (20) NOT NULL,
DATA_P DATE,
CONSTRAINT FK1 FOREIGN KEY(ID_CL) REFERENCES CLIENTE (N_TEL) ON DELETE CASCADE ,
CONSTRAINT FK2 FOREIGN KEY (ID_RISTORANTE) REFERENCES RISTORANTE(P_IVA) ON DELETE
CASCADE,
CONSTRAINT CHECK_DATA CHECK(DATA_P > DATA_CHIAMATA),
```

```
CONSTRAINT CHECK_PRENOT CHECK(TO_CHAR (DATA_P,'HH24')BETWEEN 12 AND 23 ),
```

```
CONSTRAINT CHECK_NUMERO_PERSONE CHECK(N_PERSONE <= 15) );
```

## Staff

La tabella staff è popolata dal proprietario della catena. Al fine di evitare errori di inserimento sono necessari i seguenti constraint: il formato del telefono che non prevede caratteri, il ruolo deve essere definito sul seguente insieme {FATTORINO, CASSIERE, INSERVIENTE, LAVAPIATTI, CHEF, CAMERIERE}, il salario non può essere inferiore a 850 euro.

```
CREATE TABLE STAFF(
```

```
  N_CARTELLINO NUMBER(2,0)PRIMARY KEY,
```

```
  NOME_S VARCHAR (10) NOT NULL,
```

```
  COGNOME_S VARCHAR (20) NOT NULL,
```

```
  N_TELEFONO VARCHAR (11),
```

```
  SALARIO NUMBER(4,0),
```

```
  RUOLO VARCHAR (15) NOT NULL,
```

```
  CONSTRAINT TELEFONO_STAF_CHECKCHECK(REGEXP_LIKE ( N_TELEFONO, '^([0-9]){3}+$' ) ),
```

```
  CONSTRAINT CHECK_RUOLO CHECK(REGEXP_LIKE  
(RUOLO,'FATTORINO|CASSIERE|INSERVIENTE|LAVAPIATTI|CHEF|CAMERIERE') ),
```

```
  CONSTRAINT CHECK_SALARIO CHECK(SALARIO>850) );
```

## Lavora

La tabella lavora è popolata dal proprietario per stabilire quale dipendente lavora per quale ristorante e che turni copre.

```
CREATE TABLE LAVORA(
```

```
  D_INIZIO_TURNO DATE,
```

```
  D_FINE_TURNO DATE,
```

```
  ID_RIST VARCHAR (20) NOT NULL,
```

```
  ID_STAFF NUMBER(2) NOT NULL,
```

```
  CONSTRAINT FK3 FOREIGN KEY (ID_RIST)REFERENCES RISTORANTE (P_IVA) ON DELETE CASCADE ,
```

```
  CONSTRAINT FK4 FOREIGN KEY (ID_STAFF)REFERENCES STAFF (N_CARTELLINO) ON DELETE CASCADE );
```

## Journey

La tabella Journey è popolata in primo luogo dal fattorino, il quale aggiunge le informazioni relative alla consegna che sta per effettuare ed è aggiornata e completata quando ritorna. L'unico constraint necessario per rispettare la logica di inserimento è il controllo sulle date inserite.

```
CREATE TABLE JOURNEY (
```

```
  KM_PERCORSI NUMBER (3,0),
```

```
  TIPO_VEICOLO VARCHAR (10),
```

```
  N_ORDINI_CONSEGNATI NUMBER(2,0),
```



```

D_PARTENZA DATE NOT NULL,
D_RITORNO DATE ,
ID_JOURNEY NUMBER(3,0),
ID_STAFF NUMBER (2,0),
CONSTRAINT FK5 PRIMARY KEY(ID_JOURNEY,ID_STAFF),
CONSTRAINT FK6 FOREIGN KEY (ID_STAFF) REFERENCES STAFF(N_CARTELLINO) ON DELETE CASCADE ,
CONSTRAINT D_CHECKCHECK(D_RITORNO > D_PARTENZA );

```

## Ordine

La tabella ordine è popolata dal cameriere nel caso in cui si sta effettuando un ordine per la consumazione in sala, altrimenti tale tabella è popolata dall'utente nel caso in cui si tratta di un ordine da asporto. Ordine è la tabella madre che si specializza in ordine\_casa e ordine\_sala

```

CREATE TABLE ORDINE (
    N_ORDINE NUMBER (3,0) PRIMARY KEY,
    ID_PRENOTAZIONE NUMBER(3,0) NOT NULL UNIQUE,
    CONSTRAINT FK7 FOREIGN KEY (ID_PRENOTAZIONE) REFERENCES PRENOTAZIONE (ID_PRENOTAZIONE) ON DELETE CASCADE
);

```

## Ordine casa

```

CREATE TABLE ORDINE_CASA (
    N_ORDINE NUMBER (3,0) PRIMARY KEY,
    CONSTRAINT FK8 FOREIGN KEY (N_ORDINE) REFERENCES ORDINE (N_ORDINE) ON DELETE CASCADE );

```

## Tavolo

Tavolo è popolata dal proprietario in fase di allestimento o aggiornamento del locale.

```

CREATE TABLE TAVOLO (
    N_TAVOLO NUMBER (2,0),
    CAPIENZA NUMBER (2,0) NOT NULL,
    ID_RISTORANTE VARCHAR (20),
    CONSTRAINT FK11 PRIMARY KEY (N_TAVOLO,ID_RISTORANTE),
    CONSTRAINT FK12 FOREIGN KEY (ID_RISTORANTE) REFERENCES RISTORANTE (P_IVA) ON DELETE CASCADE );

```

## Ordine sala

Nella tabella ordine sala non è stato reso obbligatorio l'attributo id\_tavolo altrimenti andrebbe in contrasto con la politica di cancellazione di un tavolo, perché in ogni caso potrebbe essere utile ricordare l'ordine effettuato anche se il tavolo venisse per un qualche motivo cancellato. Un tavolo viene assegnato ad un ordine attraverso la procedura "assegna\_tavolo()" richiamata da un cameriere.

```

CREATE TABLE ORDINE_SALA (
    N_ORDINE NUMBER (3,0) PRIMARY KEY,
    ID_TAVOLO NUMBER (2,0) ,
    ID_RISTORANTE VARCHAR (20) NOT NULL,

```

```
CONSTRAINT FK9 FOREIGN KEY (N_ORDINE) REFERENCES ORDINE (N_ORDINE) ON DELETE CASCADE ,  
  
CONSTRAINT FK10 FOREIGN KEY (ID_TAVOLO,ID_RISTORANTE) REFERENCES TAVOLO (N_TAVOLO,ID_RISTORANTE) ON  
DELETE SET NULL );
```

## Consegna

La tabella consegna è popolata dal fattorino nel momento in cui consegna il pasto d'asporto al cliente.

```
CREATE TABLE CONSEGNA  
  
SERIALE VARCHAR (10) PRIMARY KEY,  
  
ID_ORDINE NUMBER (3,0) NOT NULL UNIQUE,  
  
ID_JOURNEY NUMBER(3,0) NOT NULL,  
  
ID_STAFF NUMBER(2,0)NOT NULL,  
  
D_CONSEGNA DATE,  
  
CONSTRAINT FK13 FOREIGN KEY (ID_ORDINE) REFERENCES ORDINE_CASA(N_ORDINE) ON DELETE SET NULL,  
  
CONSTRAINT FK14 FOREIGN KEY (ID_JOURNEY,ID_STAFF) REFERENCES JOURNEY (ID_JOURNEY,ID_STAFF) ON DELETE SET NULL  
);
```

## Carta di credito

Tale tabella viene popolata dal cassiere nel momento in cui il cliente deve effettuare il pagamento del conto, o dal cliente stesso se si tratta di un ordine d'asporto.

```
CREATE TABLE CARTA_DI_CREDITO (  
  
N_CARD VARCHAR (16) PRIMARY KEY,  
  
SCADENZA DATE NOT NULL,  
  
NOME_TITOLARE VARCHAR (20) NOT NULL,  
  
COGNOME_TITOLARE VARCHAR (20) NOT NULL,  
  
ID_CLIENTE VARCHAR (11) NOT NULL,  
  
CCV CHAR (3) NOT NULL,  
  
CONSTRAINT FK15 FOREIGN KEY(ID_CLIENTE)REFERENCES CLIENTE (N_TEL) ON DELETE CASCADE );
```

## Conto

È popolata dal cassiere. Grazie alla procedura "get\_conto()" viene determinato il valore dell'attributo totale.

```
CREATE TABLE CONTO (  
  
CODICE NUMBER(4,0) PRIMARY KEY,  
  
ID_ORDINE NUMBER (3,0) NOT NULL UNIQUE,  
  
ID_CC VARCHAR (16),  
  
TOTALE NUMBER (4,0),  
  
CONSTRAINT FK16 FOREIGN KEY (ID_ORDINE) REFERENCES ORDINE(N_ORDINE) ON DELETE CASCADE ,  
  
CONSTRAINT FK17 FOREIGN KEY (ID_CC) REFERENCES CARTA_DI_CREDITO (N_CARD) ON DELETE SET NULL );
```

## Menu

I menu essendo prefissati sono caricati dal proprietario.

```
CREATE TABLE MENU(  
    ID_MENU NUMBER (2,0) PRIMARY KEY,  
    TIPO VARCHAR (20) );
```

## Selezione

Tale tabella associa un ordine a uno o più menu, attraverso l'attributo quantità determiniamo quanti menu di quello stesso tipo sono richiesti. Il popolamento della suddetta tabella è effettuato dalla procedura "seleziona\_menu()" richiamata sia dal chiamare per ordini in sala sia dall'utente per ordini d'asporto.

```
CREATE TABLE SELEZIONA (  
    QUANTITA NUMBER(2,0) NOT NULL,  
    N_ORDINE NUMBER (3,0) NOT NULL,  
    ID_MENU NUMBER (2,0) NOT NULL,  
    CONSTRAINT FK18 FOREIGN KEY (N_ORDINE) REFERENCES ORDINE (N_ORDINE) ON DELETE CASCADE ,  
    CONSTRAINT FK19 FOREIGN KEY (ID_MENU) REFERENCES MENU (ID_MENU) ON DELETE CASCADE );
```

## Pasto

La tabella pasto è popolata dal proprietario.

```
CREATE TABLE PASTO (  
    NOME_PASTO VARCHAR (30) PRIMARY KEY,  
    PREZZO NUMBER(2,0) NOT NULL  
);
```

## Contiene

Tale tabella è gestita interamente da un utente di tipo chef.

```
CREATE TABLE CONTIENE(  
    ID_MENU NUMBER(2,0) NOT NULL,  
    NOME_PASTO VARCHAR (30) NOT NULL,  
    CONSTRAINT FK20 FOREIGN KEY(ID_MENU) REFERENCES MENU(ID_MENU) ON DELETE CASCADE ,  
    CONSTRAINT FK27 FOREIGN KEY(NOME_PASTO) REFERENCES PASTO(NOME_PASTO)ON DELETE CASCADE );
```

## Ingrediente

Il proprietario è colui che carica gli ingredienti acquistati

```
CREATE TABLE INGREDIENTE (  
    BARCODE VARCHAR (8) PRIMARY KEY,
```

```
NOME_INGREDIENTE VARCHAR (30) NOT NULL );
```

## Si compone

La tabella si compone è gestita interamente da un utente di tipo chef , in quanto lui conosce gli ingredienti di ogni pasto.

```
CREATE TABLE SI_COMPONE (  
    NOME_PASTO VARCHAR (30) NOT NULL,  
    BARCODE VARCHAR (8) NOT NULL,  
    QUANTITA VARCHAR (6),  
    CONSTRAINT FK21 FOREIGN KEY(BARCODE) REFERENCES INGREDIENTE(BARCODE) ON DELETE CASCADE ,  
    CONSTRAINT FK22 FOREIGN KEY (NOME_PASTO)REFERENCES PASTO (NOME_PASTO)ON DELETE CASCADE );
```

## Fornitore

È il proprietario della catena che si occupa di memorizzare i suoi fornitori. Anche qui come nella tabella ristorante è necessario controllare i formati della partita iva e del numero di telefono.

```
CREATE TABLE FORNITORE (  
    P_IVA_FORNITORE VARCHAR (20) PRIMARY KEY,  
    NOME_FORNITORE VARCHAR (30) NOT NULL,  
    TELEFONO_FORNITORE VARCHAR (11) NOT NULL,  
    CITTA VARCHAR (15),  
    CAP VARCHAR (6),  
    VIA VARCHAR (30),  
    CONSTRAINT telefono_check_fornitore CHECK( REGEXP_LIKE (TELEFONO_FORNITORE ,'^[0-9]{3}+$' ) ),  
    CONSTRAINT CHECK_IVA_FORNITORE CHECK(REGEXP_LIKE(P_IVA_FORNITORE, '[A-Z]{2}+[A-Z0-9]{9}' ) ) );
```

## Fornitura

Il proprietario carica i dati dello scarico merci.

```
CREATE TABLE FORNITURA (  
    N_FATTURA NUMBER (3,0) PRIMARY KEY,  
    TOTALE NUMBER(4,0) NOT NULL,  
    DATA_CONSEGNA DATE,  
    ID_FORNITORE VARCHAR (20),  
    ID_RISTORANTE VARCHAR (20) NOT NULL,  
    DATA_RICHIESTA DATE,  
    CONSTRAINT FK23 FOREIGN KEY (ID_FORNITORE) REFERENCES FORNITORE (P_IVA_FORNITORE) ON DELETE SET NULL,
```

CONSTRAINT FK24 FOREIGN KEY(ID\_RISTORANTE ) REFERENCES RISTORANTE (P\_IVA) ON DELETE CASCADE );

## Proviene da

Popolata dal proprietario.

```
CREATE TABLE PROVIENE_DA (  
    BARCODE VARCHAR (8) NOT NULL,  
    N_FATTURA NUMBER (3,0) NOT NULL,  
    QUANTITA VARCHAR (6),  
    CONSTRAINT FK25 FOREIGN KEY(BARCODE)REFERENCES INGREDIENTE (BARCODE)ON DELETE CASCADE ,  
    CONSTRAINT FK26 FOREIGN KEY (N_FATTURA) REFERENCES FORNITURA (N_FATTURA) ON DELETE CASCADE );
```

## Data Manipulation Language

### Cliente

```
INSERT INTO CLIENTE VALUES ('MATTIA','ESPOSITO', '333158543','Capodichino',10,'Mattia63@gmail.com ','10122');  
INSERT INTO CLIENTE VALUES ('STEFANO','TRAPANI', '339758403','secondigliano',11,'stefano_trapani@outlook.it ','10149');  
INSERT INTO CLIENTE VALUES ('LUCA','RUBINO', '328741025','Caserta',12,'lucarubino99@gmail.com ','10138');  
INSERT INTO CLIENTE VALUES ('ANTONIO','DEL GIUDICE', '339854100','garibaldi',31,'antonio1954@gmail.com ','10124');  
INSERT INTO CLIENTE VALUES ('VITTORIO','PISCITELLI', '333368518','nuovo corso',36,'vit_pisc_li99@outlook.com ','20122');  
INSERT INTO CLIENTE VALUES ('SALVATORE','MANCINI', '320235471','del trionfo',44,'sasi_manc69@gmail.com ','20130');  
INSERT INTO CLIENTE VALUES ('ANDREA','TESTA', '335482669','della giustizia',56,'mr_head1976@gmail.com ','20155');  
INSERT INTO CLIENTE VALUES ('GIULIO','COLUCCI', '337458008','padova',69,'Colucci89@libero.it ','20133');  
INSERT INTO CLIENTE VALUES ('GIUSY','BARRUCCI', '339575315','condotti',9,'barruccGIUSY@libero.it ','00101');  
INSERT INTO CLIENTE VALUES ('STEFANIA','LA COMMARA', '334206595','del babbuino',244,'STEFANIAla_COMM@gmail.com ','00199');  
INSERT INTO CLIENTE VALUES ('CARMELA','ESPOSITO', '344854466','nuovo corso',178,'Carmela@gmail.com ','00150');  
INSERT INTO CLIENTE VALUES ('PAOLA','RUGGIERO', '315760108','del letto',7,'RUGG_PAOLA@outlook.it ','00198');  
INSERT INTO CLIENTE VALUES ('LUDOVICA','VERDE', '339514233','nuova marano',8,'ludo_1996@gmail.com ','80120');  
INSERT INTO CLIENTE VALUES ('ROBERTA','BIONDI', '340187599','petrarca',1,'roberta_biondi-1978@gmail.com ','80102');  
INSERT INTO CLIENTE VALUES ('GIUSEPPE','ROSSI', '338431865','dante',871,'peppe_red1957@outlook.com ','80144');  
INSERT INTO CLIENTE VALUES ('PASQUALE','DI LORENZO', '333124567','cimitero',718,'il_paquito1988@libero.it ','80140');  
INSERT INTO CLIENTE VALUES ('MARIO','OLIVA', '385701694','capodichino',243,'mario.oliva243@gmail.com ','20125');  
INSERT INTO CLIENTE VALUES ('RENATO','ESPOSITO', '339140509','fosso del lupo',9,'renato.esposito1999@outlook.com ','20155');
```

### Ristorante

```
INSERT INTO RISTORANTE VALUES ('IT123456789','LaBaia','Napoleone','10121','Torino',5,'1');  
INSERT INTO RISTORANTE VALUES ('IT987654321','LaBaia','Badante','20150','Milano',18,'2');  
INSERT INTO RISTORANTE VALUES ('IT237858945','LaBaia','Europa','00013','Roma',10,'3');  
INSERT INTO RISTORANTE VALUES ('IT789415263','LaBaia','Calata Capodichino','80170','Napoli',23,'1');
```

### Prenotazione

INSERT INTO PRENOTAZIONE VALUES (1,5,'333158543',TO\_DATE ('04-GIU-2020  
11:30 ','DD-MON -YYYY HH24 :MI') , 'IT123456789' ,TO\_DATE ('07-GIU-2020  
15:30 ','DD-MON-YYYY HH24 :MI'));

INSERT INTO PRENOTAZIONE VALUES (2,10,'339758403',TO\_DATE ('28-MAG-2020  
14:33 ','DD -MON - YYYY HH24 :MI ') , 'IT123456789' ,TO\_DATE ('28-MAG-2020  
19:00 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (3,15,'328741025',TO\_DATE ('30-MAG-2020  
18:50 ','DD -MON - YYYY HH24 :MI ') , 'IT123456789' ,TO\_DATE ('31-MAG-2020  
20:45 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (4,8,'339854100',TO\_DATE ('31-MAG -2020  
17:45 ','DD -MON - YYYY HH24 :MI ') , 'IT123456789' ,TO\_DATE ('31-MAG-2020  
21:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (5,2,'333368518',TO\_DATE ('03-GIU-2020  
13:36 ','DD -MON - YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('03-GIU-2020  
22:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (6,4,'320235471',TO\_DATE ('01-GIU -2020  
10:44 ','DD -MON - YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('01-GIU-2020  
12:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (7,2,'335482669',TO\_DATE ('27-MAG -2020  
17:30 ','DD -MON - YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('29-MAG-2020  
21:00 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (8,8,'337458008',TO\_DATE ('25-MAG -2020  
18:22 ','DD -MON - YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('1-GIU-2020  
19:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (9,15,'339575315',TO\_DATE ('1-GIU-2020  
11:21 ','DD -MON - YYYY HH24 :MI ') , 'IT237858945' ,TO\_DATE ('2-GIU-2020  
14:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (10,15,'334206595',TO\_DATE ('1-GIU-2020  
20:00 ','DD -MON - YYYY HH24 :MI ') , 'IT237858945' ,TO\_DATE ('1-GIU-2020  
23:00 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (11,3,'344854466',TO\_DATE ('2-GIU-2020  
10:56 ','DD -MON - YYYY HH24 :MI ') , 'IT237858945' ,TO\_DATE ('2-GIU-2020  
13:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (12,4,'315760108',TO\_DATE ('1-GIU-2020  
11:23 ','DD -MON - YYYY HH24 :MI ') , 'IT237858945' ,TO\_DATE ('4-GIU-2020  
14:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (13,8,'339514233',TO\_DATE ('4-GIU-2020  
15:17 ','DD -MON - YYYY HH24 :MI ') , 'IT789415263' ,TO\_DATE ('4-GIU-2020  
23:00 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (14,2,'340187599',TO\_DATE ('31-MAG -2020  
16:16 ','DD -MON - YYYY HH24 :MI ') , 'IT789415263' ,TO\_DATE ('2-GIU-2020  
19:45 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (15,5,'338431865',TO\_DATE ('28-MAG -2020  
15:15 ','DD -MON - YYYY HH24 :MI ') , 'IT789415263' ,TO\_DATE ('30-MAG-2020  
20:15 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (16,1,'333124567',TO\_DATE ('31-MAG-2020  
16:40 ','DD-MON-YYYY HH24 :MI ') , 'IT789415263' ,TO\_DATE ('31-MAG-2020  
21:45 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (17,1,'385701694',TO\_DATE ('03-GIU-2020  
16:29 ','DD-MON-YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('03-GIU-2020  
19:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (18,1,'339140509',TO\_DATE ('31-MAG -2020  
10:30 ','DD-MON-YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('31-MAG -2020  
13:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (19,1,'340187599',TO\_DATE ('27-MAG -2020  
10:30 ','DD-MON-YYYY HH24 :MI ') , 'IT789415263' ,TO\_DATE ('27-MAG -2020  
13:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (20,1,'328741025',TO\_DATE ('06-GIU -2020  
10:30 ','DD-MON-YYYY HH24 :MI ') , 'IT123456789' ,TO\_DATE ('06-GIU -2020  
20:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (21,1,'334206595',TO\_DATE ('07-GIU -2020  
12:30 ','DD-MON-YYYY HH24 :MI ') , 'IT237858945' ,TO\_DATE ('07-GIU -2020  
21:15 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (22,1,'339758403',TO\_DATE ('02-GIU-2020  
12:15 ','DD -MON - YYYY HH24 :MI ') , 'IT123456789' ,TO\_DATE ('02-GIU -2020  
14:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (23,1,'339140509',TO\_DATE ('28-MAG -2020  
15:30 ','DD -MON - YYYY HH24 :MI ') , 'IT987654321' ,TO\_DATE ('28-MAG -2020  
20:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (24,1,'339575315',TO\_DATE ('11-GIU -2020  
11:30 ','DD -MON - YYYY HH24 :MI ') , 'IT237858945' ,TO\_DATE ('11-GIU -2020  
13:25 ','DD -MON - YYYY HH24 :MI '));

```

INSERT INTO PRENOTAZIONE VALUES (25,1,'338431865',TO_DATE ('27-MAG -2020
11:45 ','DD -MON - YYYY HH24 :MI ') , 'IT789415263' ,TO_DATE ('27-MAG -2020
12:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (26,1,'339854100',TO_DATE ('22-MAG -2020
11:40 ','DD -MON - YYYY HH24 :MI ') , 'IT123456789' ,TO_DATE ('22-MAG -2020
19:45 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (27,1,'333158543',TO_DATE ('25-MAG -2020
17:30 ','DD -MON - YYYY HH24 :MI ') , 'IT237858945' ,TO_DATE ('25-MAG -2020
20:30 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (28,1,'385701694',TO_DATE ('04-GIU-2020
12:30 ','DD-MON-YYYY HH24 :MI ') , 'IT987654321' ,TO_DATE ('04-GIU-2020
14:45 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (29,1,'340187599',TO_DATE ('03-GIU-2020
15:00 ','DD-MON-YYYY HH24 :MI ') , 'IT789415263' ,TO_DATE ('03-GIU-2020
22:15 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (30,1,'328741025',TO_DATE ('02-GIU-2020
11:30 ','DD-MON-YYYY HH24 :MI ') , 'IT123456789' ,TO_DATE ('02-GIU-2020
15:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (31,1,'340187599',TO_DATE ('31-MAG -2020
14:30 ','DD-MON-YYYY HH24 :MI ') , 'IT789415263' ,TO_DATE ('31-MAG -2020
20:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (32,1,'339575315',TO_DATE ('28-MAG -2020
12:15 ','DD-MON-YYYY HH24 :MI ') , 'IT237858945' ,TO_DATE ('28-MAG -2020
13:30 ','DD-MON-YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (33,1,'335482669',TO_DATE ('10-GIU -2020
13:30 ','DD -MON - YYYY HH24 :MI ') , 'IT987654321' ,TO_DATE ('10-GIU -2020
14:45 ','DD -MON - YYYY HH24 :MI '));

INSERT INTO PRENOTAZIONE VALUES (34,1,'333124567',TO_DATE ('09-GIU -2020
12:30 ','DD -MON - YYYY HH24 :MI ') , 'IT789415263' ,TO_DATE ('09-GIU-2020
21:00 ','DD -MON - YYYY HH24 :MI '));

```

## Staff

```

INSERT INTO STAFF VALUES (1,'GIULIANO','SAGESE','033125590',1250,'INSERVIENTE');
INSERT INTO STAFF VALUES (2,'FABRIZIA','PISANI','038457386',1350,'INSERVIENTE');
INSERT INTO STAFF VALUES (3,'FABIO','CAPELLO','038254982',1100,'INSERVIENTE');

```



INSERT INTO STAFF VALUES (4,'GENNARO','GATTUSO','036485291',1310,'INSERVIENTE');

INSERT INTO STAFF VALUES (5,'FEDERICA','DI BENEDETTO','320541698',1180,'INSERVIENTE');

INSERT INTO STAFF VALUES (6,'EMANUELA','CIRILLO','320174123',1410,'INSERVIENTE');

INSERT INTO STAFF VALUES (7,'STEFANIA','TEDESCHI','333123458',1275,'INSERVIENTE');

INSERT INTO STAFF VALUES (8,'MIRCO','ROMANO','330147893',1150,'INSERVIENTE');

INSERT INTO STAFF VALUES (9,'CRISTIAN','PISCITELLI','330456852',1250,'FATTORINO');

INSERT INTO STAFF VALUES (10,'TERESA','SAVARESE','330753159',1232,'FATTORINO');

INSERT INTO STAFF VALUES (11,'VALERIA','NANI','327005873',1230,'FATTORINO');

INSERT INTO STAFF VALUES (12,'FRANCESCA','FRANZESE','309221818',1180,'FATTORINO');

INSERT INTO STAFF VALUES (13,'ALESSIO','COPPOLA','325461722',1165,'FATTORINO');

INSERT INTO STAFF VALUES (14,'GIOVANNI','MICELI','333121452',1200,'FATTORINO');

INSERT INTO STAFF VALUES (15,'GUIDO','MURA','389211467',1190,'FATTORINO');

INSERT INTO STAFF VALUES (16,'GIADA','CRISTIANO','331741291',1180,'FATTORINO');

INSERT INTO STAFF VALUES (17,'FRANCO','SORBILLO','319899219',1242,'FATTORINO');

INSERT INTO STAFF VALUES (18,'CIRO','PICCOLO','333325471',1240,'FATTORINO');

INSERT INTO STAFF VALUES (19,'PAOLA','FRANCHI','322123422',1050,'FATTORINO');

INSERT INTO STAFF VALUES (20,'STEFANIA','SARDU','325564560',1121,'FATTORINO');

INSERT INTO STAFF VALUES (21,'ALESSANDRA','DALLA','309127092',1136,'FATTORINO');

INSERT INTO STAFF VALUES (22,'MICHELE','DANIELE','333441456',1200,'FATTORINO');

INSERT INTO STAFF VALUES (23,'GIOVANNI','STIRPIO','331245432',1222,'FATTORINO');

INSERT INTO STAFF VALUES (24,'GIOVANNA','TRAPANI','355655789',1233,'FATTORINO');

INSERT INTO STAFF VALUES (25,'ROSA','PAGANI','333225641',1212,'LAVAPIATTI');

INSERT INTO STAFF VALUES (26,'DARIO','CASTELLANO','322241156',1110,'LAVAPIATTI');

INSERT INTO STAFF VALUES (27,'GENNARO','SECCE','333237100',1171,'LAVAPIATTI');

INSERT INTO STAFF VALUES (28,'MATTIA','DI MAURO','353556557',1050,'LAVAPIATTI');

INSERT INTO STAFF VALUES (29,'PASQUALINO','ACAMPORA','333342442',1000,'LAVAPIATTI');

INSERT INTO STAFF VALUES (30,'LUIGI','SEGREGO','345336271',1015,'LAVAPIATTI');

INSERT INTO STAFF VALUES (31,'CHIARA','ROSSI','333112452',1101,'LAVAPIATTI');

INSERT INTO STAFF VALUES (32,'ROSSELLA','COGLIO','433526671',1136,'LAVAPIATTI');

INSERT INTO STAFF VALUES (33,'RAFFAELE','FUMO','351142311',1222,'CASSIERE');

INSERT INTO STAFF VALUES (34,'DAVIDE','GINOLA','335277689',1201,'CASSIERE');

INSERT INTO STAFF VALUES (35,'ADRIANO','CALABRESE','344562212',1000,'CASSIERE');

INSERT INTO STAFF VALUES (36,'CHIARA','STANIO','399098800',1121,'CASSIERE');

INSERT INTO STAFF VALUES (37,'PIETRO','MORATI','334119765',1010,'CASSIERE');

INSERT INTO STAFF VALUES (38,'CARMELA','STAIANO','459909890',1014,'CASSIERE');

INSERT INTO STAFF VALUES (39,'PAOLA','BRUCHI','566655781',1032,'CASSIERE');

INSERT INTO STAFF VALUES (40,'VANESSA','DI NAPOLI','661188987',1213,'CASSIERE');

INSERT INTO STAFF VALUES (41,'RAFFAELA','STAMO','223909800',2214,'CAMERIERE');

INSERT INTO STAFF VALUES (42,'STEFANO','MARINO','442156234',2100,'CAMERIERE');

```

INSERT INTO STAFF VALUES (43,'PIETRO','ROMANO','556223451',2050,'CAMERIERE');
INSERT INTO STAFF VALUES (44,'SIMONA','CONTI','665378910',2250,'CAMERIERE');
INSERT INTO STAFF VALUES (45,'ALESSANDRA','RIZZO','336627189',2150,'CAMERIERE');
INSERT INTO STAFF VALUES (46,'SALVATORE','ESPOSITO','446321900',2000,'CAMERIERE');
INSERT INTO STAFF VALUES (47,'VALERIO','MORETTI','333356555',2110,'CAMERIERE');
INSERT INTO STAFF VALUES (48,'GIUSEPPE','PENSA','478233165',2209,'CAMERIERE');
INSERT INTO STAFF VALUES (49,'LINO','BARBIERI','445521678',2190,'CAMERIERE');
INSERT INTO STAFF VALUES (50,'FABRIZIO','FONTANA','612133452',2140,'CAMERIERE');
INSERT INTO STAFF VALUES (51,'CHIARA','RINALDI','333311221',2005,'CAMERIERE');
INSERT INTO STAFF VALUES (52,'FIONA','GRECO','334451217',2015,'CAMERIERE');
INSERT INTO STAFF VALUES (53,'LUCA','GALLO','717890121',2230,'CAMERIERE');
INSERT INTO STAFF VALUES (54,'UMBERTO','MARIANI','561172314',2235,'CAMERIERE');
INSERT INTO STAFF VALUES (55,'SAVIO','CARUSO','432334781',2300,'CAMERIERE');
INSERT INTO STAFF VALUES (56,'CATERINA','FERRARI','356177556',2290,'CAMERIERE');
INSERT INTO STAFF VALUES (57,'DIEGO','FERRARA','309155323',3130,'CHEF');
INSERT INTO STAFF VALUES (58,'SARA','COSTA','330098012',3250,'CHEF');
INSERT INTO STAFF VALUES (59,'DOMENICO','VITALE','288283321',3110,'CHEF');
INSERT INTO STAFF VALUES (60,'ACHILLE','DE SANTIS','223167509',3000,'CHEF');
INSERT INTO STAFF VALUES (61,'MARCO','LOMBARDO','717231158',3220,'CHEF');
INSERT INTO STAFF VALUES (62,'MELISSA','MONTI','330010293',3000,'CHEF');
INSERT INTO STAFF VALUES (63,'VALENTINA','RIZZI','459009211',3220,'CHEF');
INSERT INTO STAFF VALUES (64,'VALERIO','FARINA','331332109',3030,'CHEF');

```

## Lavora

```

INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT123456789', 1 );
INSERT INTO LAVORA VALUES (TO_DATE ('18:00 ','HH24 :MI '), TO_DATE ('23:30 ','HH24 :MI '), 'IT123456789', 2 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT987654321', 3 );
INSERT INTO LAVORA VALUES (TO_DATE ('18:00 ','HH24 :MI '), TO_DATE ('23:30 ','HH24 :MI '), 'IT987654321', 4 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT237858945', 5 );
INSERT INTO LAVORA VALUES (TO_DATE ('18:00 ','HH24 :MI '), TO_DATE ('23:30 ','HH24 :MI '), 'IT237858945', 6 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT789415263', 7 );
INSERT INTO LAVORA VALUES (TO_DATE ('18:00 ','HH24 :MI '), TO_DATE ('23:30 ','HH24 :MI '), 'IT789415263', 8 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT123456789', 9 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT123456789', 10 );
INSERT INTO LAVORA VALUES (TO_DATE ('18:00 ','HH24 :MI '), TO_DATE ('23:30 ','HH24 :MI '), 'IT123456789', 11 );
INSERT INTO LAVORA VALUES (TO_DATE ('18:00 ','HH24 :MI '), TO_DATE ('23:30 ','HH24 :MI '), 'IT123456789', 12 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT987654321', 13 );
INSERT INTO LAVORA VALUES (TO_DATE ('11:30 ','HH24 :MI '), TO_DATE ('17:30 ','HH24 :MI '), 'IT987654321', 14 );

```

[illegible]

```

INSERT INTO LAVORA VALUES(TO_DATE('11:30','HH24 :MI'), TO_DATE('17:30','HH24 :MI'), 'IT789415263', 54);
INSERT INTO LAVORA VALUES(TO_DATE('18:00','HH24 :MI'), TO_DATE('23:30','HH24 :MI'), 'IT789415263', 55);
INSERT INTO LAVORA VALUES(TO_DATE('18:00','HH24 :MI'), TO_DATE('23:30','HH24 :MI'), 'IT789415263', 56);
INSERT INTO LAVORA VALUES(TO_DATE('11:30','HH24 :MI'), TO_DATE('17:30','HH24 :MI'), 'IT123456789', 57);
INSERT INTO LAVORA VALUES(TO_DATE('18:00','HH24 :MI'), TO_DATE('23:30','HH24 :MI'), 'IT123456789', 58);
INSERT INTO LAVORA VALUES(TO_DATE('11:30','HH24 :MI'), TO_DATE('17:30','HH24 :MI'), 'IT987654321', 59);
INSERT INTO LAVORA VALUES(TO_DATE('18:00','HH24 :MI'), TO_DATE('23:30','HH24 :MI'), 'IT987654321', 60);
INSERT INTO LAVORA VALUES(TO_DATE('11:30','HH24 :MI'), TO_DATE('17:30','HH24 :MI'), 'IT237858945', 61);
INSERT INTO LAVORA VALUES(TO_DATE('18:00','HH24 :MI'), TO_DATE('23:30','HH24 :MI'), 'IT237858945', 62);
INSERT INTO LAVORA VALUES(TO_DATE('11:30','HH24 :MI'), TO_DATE('17:30','HH24 :MI'), 'IT789415263', 63);
INSERT INTO LAVORA VALUES(TO_DATE('18:00','HH24 :MI'), TO_DATE('23:30','HH24 :MI'), 'IT789415263', 64);

```

## Journey

```

INSERT INTO JOURNEY VALUES(13, 'MACCHINA', 1, TO_DATE ('31-MAG-2020
21:30 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('31-MAG-2020
22:00 ','DD -MON - YYYY HH24 :MI '), 1, 24);
INSERT INTO JOURNEY VALUES(14, 'MACCHINA', 1, TO_DATE ('03-GIU-2020
19:20 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('03-GIU-2020
19:45 ','DD -MON - YYYY HH24 :MI '), 2, 15);
INSERT INTO JOURNEY VALUES(12, 'MACCHINA', 1, TO_DATE ('31-MAG-2020
13:20 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('31-MAG-2020
13:45 ','DD -MON - YYYY HH24 :MI '), 3, 22);
INSERT INTO JOURNEY VALUES(10, 'MACCHINA', 1, TO_DATE ('27-MAG-2020
13:20 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('27-MAG-2020
13:45 ','DD -MON - YYYY HH24 :MI '), 4, 14);
INSERT INTO JOURNEY VALUES(7, 'MACCHINA', 1, TO_DATE ('06-GIU-2020
20:15 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('06-GIU-2020
20:45 ','DD -MON - YYYY HH24 :MI '), 5, 11);
INSERT INTO JOURNEY VALUES(7, 'MACCHINA', 1, TO_DATE ('07-GIU-2020
21:00 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('07-GIU-2020
21:30 ','DD -MON - YYYY HH24 :MI '), 6, 19);
INSERT INTO JOURNEY VALUES(8, 'MACCHINA', 1, TO_DATE ('02-GIU-2020
14:10 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('02-GIU-2020
14:45 ','DD -MON - YYYY HH24 :MI '), 7, 9);
INSERT INTO JOURNEY VALUES(12, 'MACCHINA', 1, TO_DATE ('28-MAG-2020
20:15 ','DD -MON - YYYY HH24 :MI '), TO_DATE ('28-MAG-2020
20:45 ','DD -MON - YYYY HH24 :MI '), 8, 16);
INSERT INTO JOURNEY VALUES(14, 'MACCHINA', 1, TO_DATE ('11-GIU-2020

```

13:00 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('11-GIU-2020

13:40 ','DD -MON - YYYY HH24 :MI ') , 9, 17);

INSERT INTO JOURNEY VALUES(7, 'MACCHINA', 1, TO\_DATE ('27-MAG-2020

12:00 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('27-MAG-2020

12:55 ','DD -MON - YYYY HH24 :MI ') , 10, 21);

INSERT INTO JOURNEY VALUES(11, 'MOTORINO', 1, TO\_DATE ('22-MAG-2020

19:35 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('22-MAG-2020

20:05 ','DD -MON - YYYY HH24 :MI ') , 11, 12);

INSERT INTO JOURNEY VALUES(12, 'MACCHINA', 1, TO\_DATE ('25-MAG-2020

20:30 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('25-MAG-2020

21:00 ','DD -MON - YYYY HH24 :MI ') , 12, 19);

INSERT INTO JOURNEY VALUES(12, 'MOTORINO', 1, TO\_DATE ('04-GIU-2020

14:40 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('04-GIU-2020

14:50 ','DD -MON - YYYY HH24 :MI ') , 13, 14);

INSERT INTO JOURNEY VALUES(5, 'MOTORINO', 1, TO\_DATE ('03-GIU-2020

22:00 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('03-GIU-2020

22:30 ','DD -MON - YYYY HH24 :MI ') , 14, 23);

INSERT INTO JOURNEY VALUES(15, 'MACCHINA', 1, TO\_DATE ('01-GIU-2020

15:00 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('01-GIU-2020

16:05 ','DD -MON - YYYY HH24 :MI ') , 15, 10);

INSERT INTO JOURNEY VALUES(15, 'MOTORINO', 1, TO\_DATE ('31-MAG-2020

20:05 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('31-MAG-2020

20:55 ','DD -MON - YYYY HH24 :MI ') , 16, 24);

INSERT INTO JOURNEY VALUES(10, 'MACCHINA', 1, TO\_DATE ('28-MAG-2020

13:20 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('28-MAG-2020

13:55 ','DD -MON - YYYY HH24 :MI ') , 17, 18);

INSERT INTO JOURNEY VALUES(10, 'MACCHINA', 1, TO\_DATE ('10-GIU-2020

14:30 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('10-GIU-2020

15:10 ','DD -MON - YYYY HH24 :MI ') , 18, 13);

INSERT INTO JOURNEY VALUES(10, 'MACCHINA', 1, TO\_DATE ('08-GIU-2020

20:30 ','DD -MON - YYYY HH24 :MI ') , TO\_DATE ('08-GIU-2020

21:20 ','DD -MON - YYYY HH24 :MI ') , 19, 23);

## Ordine

INSERT INTO ORDINE VALUES(101,1);

INSERT INTO ORDINE VALUES(102,2);

INSERT INTO ORDINE VALUES(103,3);

INSERT INTO ORDINE VALUES(104,4);

```
INSERT INTO ORDINE VALUES(105,5);
INSERT INTO ORDINE VALUES(106,6);
INSERT INTO ORDINE VALUES(107,7);
INSERT INTO ORDINE VALUES(108,8);
INSERT INTO ORDINE VALUES(109,9);
INSERT INTO ORDINE VALUES(110,10);
INSERT INTO ORDINE VALUES(111,11);
INSERT INTO ORDINE VALUES(112,12);
INSERT INTO ORDINE VALUES(113,13);
INSERT INTO ORDINE VALUES(114,14);
INSERT INTO ORDINE VALUES(115,15);
INSERT INTO ORDINE VALUES(116,16);
INSERT INTO ORDINE VALUES(117,17);
INSERT INTO ORDINE VALUES(118,18);
INSERT INTO ORDINE VALUES(119,19);
INSERT INTO ORDINE VALUES(120,20);
INSERT INTO ORDINE VALUES(121,21);
INSERT INTO ORDINE VALUES(122,22);
INSERT INTO ORDINE VALUES(123,23);
INSERT INTO ORDINE VALUES(124,24);
INSERT INTO ORDINE VALUES(125,25);
INSERT INTO ORDINE VALUES(126,26);
INSERT INTO ORDINE VALUES(127,27);
INSERT INTO ORDINE VALUES(128,28);
INSERT INTO ORDINE VALUES(129,29);
INSERT INTO ORDINE VALUES(130,30);
INSERT INTO ORDINE VALUES(131,31);
INSERT INTO ORDINE VALUES(132,32);
INSERT INTO ORDINE VALUES(133,33);
INSERT INTO ORDINE VALUES(134,34);
```

## Tavolo

```
INSERT INTO TAVOLO VALUES(1, 6, 'IT237858945');
INSERT INTO TAVOLO VALUES(2, 4, 'IT237858945');
INSERT INTO TAVOLO VALUES(3, 10, 'IT237858945');
INSERT INTO TAVOLO VALUES(4, 2, 'IT237858945');
INSERT INTO TAVOLO VALUES(5, 8, 'IT237858945');
INSERT INTO TAVOLO VALUES(6, 2, 'IT237858945');
INSERT INTO TAVOLO VALUES(7, 12, 'IT237858945');
INSERT INTO TAVOLO VALUES(8, 15, 'IT237858945');
```

```
INSERT INTO TAVOLO VALUES(9, 4, 'IT237858945');
INSERT INTO TAVOLO VALUES(10, 2, 'IT237858945');
INSERT INTO TAVOLO VALUES(1, 4, 'IT789415263');
INSERT INTO TAVOLO VALUES(2, 8, 'IT789415263');
INSERT INTO TAVOLO VALUES(3, 8, 'IT789415263');
INSERT INTO TAVOLO VALUES(4, 2, 'IT789415263');
INSERT INTO TAVOLO VALUES(5, 2, 'IT789415263');
INSERT INTO TAVOLO VALUES(6, 14, 'IT789415263');
INSERT INTO TAVOLO VALUES(7, 13, 'IT789415263');
INSERT INTO TAVOLO VALUES(8, 15, 'IT789415263');
INSERT INTO TAVOLO VALUES(9, 3, 'IT789415263');
INSERT INTO TAVOLO VALUES(10, 3, 'IT789415263');
INSERT INTO TAVOLO VALUES(1,4,'IT123456789');
INSERT INTO TAVOLO VALUES(2,4,'IT123456789');
INSERT INTO TAVOLO VALUES(3,6,'IT123456789');
INSERT INTO TAVOLO VALUES(4,10,'IT123456789');
INSERT INTO TAVOLO VALUES(5,15,'IT123456789');
INSERT INTO TAVOLO VALUES(6,4,'IT123456789');
INSERT INTO TAVOLO VALUES(7,4,'IT123456789');
INSERT INTO TAVOLO VALUES(8,8,'IT123456789');
INSERT INTO TAVOLO VALUES(9,12,'IT123456789');
INSERT INTO TAVOLO VALUES(10,10,'IT123456789');
```

```
INSERT INTO TAVOLO VALUES(1,4,'IT987654321');
INSERT INTO TAVOLO VALUES(2,4,'IT987654321');
INSERT INTO TAVOLO VALUES(3,6,'IT987654321');
INSERT INTO TAVOLO VALUES(4,10,'IT987654321');
INSERT INTO TAVOLO VALUES(5,15,'IT987654321');
INSERT INTO TAVOLO VALUES(6,4,'IT987654321');
INSERT INTO TAVOLO VALUES(7,4,'IT987654321');
INSERT INTO TAVOLO VALUES(8,8,'IT987654321');
INSERT INTO TAVOLO VALUES(9,12,'IT987654321');
INSERT INTO TAVOLO VALUES(10,10,'IT987654321');
```

## Ordine Sala

```
INSERT INTO ORDINE_SALA VALUES(101, 3, 'IT123456789');
INSERT INTO ORDINE_SALA VALUES(102, 10, 'IT123456789');
INSERT INTO ORDINE_SALA VALUES(103, 5, 'IT123456789');
INSERT INTO ORDINE_SALA VALUES(104, 9, 'IT123456789');
INSERT INTO ORDINE_SALA VALUES(105, 1, 'IT987654321');
```

```
INSERT INTO ORDINE_SALA VALUES(106, 2, 'IT987654321');
INSERT INTO ORDINE_SALA VALUES(107, 7, 'IT987654321');
INSERT INTO ORDINE_SALA VALUES(108, 8, 'IT987654321');
INSERT INTO ORDINE_SALA VALUES(109, 8, 'IT237858945');
INSERT INTO ORDINE_SALA VALUES(110, 8, 'IT237858945');
INSERT INTO ORDINE_SALA VALUES(111, 2, 'IT237858945');
INSERT INTO ORDINE_SALA VALUES(112, 9, 'IT237858945');
INSERT INTO ORDINE_SALA VALUES(113, 2, 'IT789415263');
INSERT INTO ORDINE_SALA VALUES(114, 4, 'IT789415263');
INSERT INTO ORDINE_SALA VALUES(115, 3, 'IT789415263');
```

## Ordine casa

```
INSERT INTO ORDINE_CASA VALUES(116);
INSERT INTO ORDINE_CASA VALUES(117);
INSERT INTO ORDINE_CASA VALUES(118);
INSERT INTO ORDINE_CASA VALUES(119);
INSERT INTO ORDINE_CASA VALUES(120);
INSERT INTO ORDINE_CASA VALUES(121);
INSERT INTO ORDINE_CASA VALUES(122);
INSERT INTO ORDINE_CASA VALUES(123);
INSERT INTO ORDINE_CASA VALUES(124);
INSERT INTO ORDINE_CASA VALUES(125);
INSERT INTO ORDINE_CASA VALUES(126);
INSERT INTO ORDINE_CASA VALUES(127);
INSERT INTO ORDINE_CASA VALUES(128);
INSERT INTO ORDINE_CASA VALUES(129);
INSERT INTO ORDINE_CASA VALUES(130);
INSERT INTO ORDINE_CASA VALUES(131);
INSERT INTO ORDINE_CASA VALUES(132);
INSERT INTO ORDINE_CASA VALUES(133);
INSERT INTO ORDINE_CASA VALUES(134);
```

## Consegna

```
INSERT INTO CONSEGNA VALUES('AA1', 116, 1, 24, TO_DATE('31-MAG-2020 21:45', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA2', 117, 2, 15, TO_DATE('03-GIU-2020 19:32', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA3', 118, 3, 22, TO_DATE('31-MAG-2020 13:32', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA4', 119, 4, 14, TO_DATE('27-MAG-2020 13:32', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA5', 120, 5, 11, TO_DATE('16-GIU-2020 20:30', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA6', 121, 6, 19, TO_DATE('07-GIU-2020 21:15', 'DD -MON - YYYY HH24 :MI'));
```



```

INSERT INTO CONSEGNA VALUES('AA7', 122, 7, 9, TO_DATE('02-GIU-2020 14:27', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA8', 123, 8, 16, TO_DATE('28-MAG-2020 20:32', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AA9', 124, 9, 17, TO_DATE('11-GIU-2020 13:20', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB1', 125, 10, 21, TO_DATE('27-MAG-2020 12:27', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB2', 126, 11, 12, TO_DATE('22-MAG-2020 19:50', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB3', 127, 12, 19, TO_DATE('25-MAG-2020 20:45', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB4', 128, 13, 14, TO_DATE('04-GIU-2020 14:45', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB5', 129, 14, 23, TO_DATE('13-GIU-2020 22:15', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB6', 130, 15, 10, TO_DATE('01-GIU-2020 15:27', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB7', 131, 16, 24, TO_DATE('31-MAG-2020 20:30', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB8', 132, 17, 18, TO_DATE('28-MAG-2020 13:37', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AB9', 133, 18, 13, TO_DATE('10-GIU-2020 14:50', 'DD -MON - YYYY HH24 :MI'));
INSERT INTO CONSEGNA VALUES('AC1', 134, 19, 23, TO_DATE('08-GIU-2020 20:55', 'DD -MON - YYYY HH24 :MI'));

```

## Carta di credito

```

INSERT INTO CARTA_DI_CREDITO VALUES('4916620184642536',TO_DATE('MAG-2024','MON-YYYY'),'MATTIA','ESPOSITO','333158543','261');
INSERT INTO CARTA_DI_CREDITO VALUES('4916808663258482',TO_DATE('APR-2023','MON-YYYY'),'STEFANO','TRAPANI','339758403','123');
INSERT INTO CARTA_DI_CREDITO VALUES('4485573546155772',TO_DATE('GEN-2023','MON-YYYY'),'LUCA','RUBINO','328741025','837');
INSERT INTO CARTA_DI_CREDITO VALUES('6011450050067868',TO_DATE('OTT-2025','MON-YYYY'),'ANTONIO','DEL GIUDICE','339854100','736');
INSERT INTO CARTA_DI_CREDITO VALUES('6011624416498112',TO_DATE('OTT-2024','MON-YYYY'),'VITTORIO','PISCITELLI','333368518','612');
INSERT INTO CARTA_DI_CREDITO VALUES('6011204046303272',TO_DATE('SET-2026','MON-YYYY'),'SALVATORE','MANCINI','320235471','126');
INSERT INTO CARTA_DI_CREDITO VALUES('3045011056435083',TO_DATE('DIC-2023','MON-YYYY'),'ANDREA','TESTA','335482669','312');
INSERT INTO CARTA_DI_CREDITO VALUES('3016599551103201',TO_DATE('GIU-2024','MON-YYYY'),'GIULIO','COLUCCI','337458008','621');
INSERT INTO CARTA_DI_CREDITO VALUES('3011512846189401',TO_DATE('NOV-2024','MON-YYYY'),'GIUSY','BARRUCCI','339575315','738');
INSERT INTO CARTA_DI_CREDITO VALUES('4175002322463297',TO_DATE('GEN-2026','MON-YYYY'),'STEFANIA','LA COMMARA','334206595','351');
INSERT INTO CARTA_DI_CREDITO VALUES('4917571538297247',TO_DATE('MAG-2023','MON-YYYY'),'CARMELA','ESPOSITO','344854466','513');
INSERT INTO CARTA_DI_CREDITO VALUES('3761280974303211',TO_DATE('MAG-2026','MON-YYYY'),'PAOLA','RUGGIERO','315760108','623');
INSERT INTO CARTA_DI_CREDITO VALUES('4917166130059358',TO_DATE('MAG-2024','MON-YYYY'),'LUDOVICA','VERDE','339514233','542');
INSERT INTO CARTA_DI_CREDITO VALUES('5163448460251351',TO_DATE('MAG-2027','MON-YYYY'),'ROBERTA','BIONDI','340187599','908');
INSERT INTO CARTA_DI_CREDITO VALUES('5313473422589580',TO_DATE('FEB-2026','MON-YYYY'),'GIUSEPPE','ROSSI','338431865','076');
INSERT INTO CARTA_DI_CREDITO VALUES('2221002749460106',TO_DATE('OTT-2027','MON-YYYY'),'PASQUALE','DI LORENZO','333124567','154');
INSERT INTO CARTA_DI_CREDITO VALUES('3542161225435386',TO_DATE('DIC-2024','MON-YYYY'),'RENATO','ESPOSITO','339140509','701');
INSERT INTO CARTA_DI_CREDITO VALUES('3537118482469013',TO_DATE('FEB-2024','MON-YYYY'),'MARIO','OLIVA','385701694','101');
INSERT INTO CARTA_DI_CREDITO VALUES('3533759527571575',TO_DATE('GIU-2023','MON-YYYY'),'PAOLA','RUGGIERO','315760108','462');
INSERT INTO CARTA_DI_CREDITO VALUES('5489135982788062',TO_DATE('AGO-2023','MON-YYYY'),'CARMELA','ESPOSITO','344854466','341');

```

```
INSERT INTO CARTA_DI_CREDITO VALUES('5433713173245402',TO_DATE('FEB-2023','MON-YYYY'),'MATTIA','ESPOSITO','333158543','511');

INSERT INTO CARTA_DI_CREDITO VALUES('5544949675041610',TO_DATE('AGO-2023','MON-YYYY'),'GIUSY','BARRUCCI','339575315','114');

INSERT INTO CARTA_DI_CREDITO VALUES('5020090854226632',TO_DATE('APR-2027','MON-YYYY'),'ANTONIO','DEL GIUDICE','339854100','912');

INSERT INTO CARTA_DI_CREDITO VALUES('5038463281597521',TO_DATE('MAR-2027','MON-YYYY'),'MATTIA','ESPOSITO','333158543','879');

INSERT INTO CARTA_DI_CREDITO VALUES('4716183910043265',TO_DATE('DIC-2027','MON-YYYY'),'ANDREA','TESTA','335482669','843');
```

## Conto

Il totale della tabella conto è reso null a scopo dimostrativo, infatti chiamando la procedura get conto è possibile aggiornare tali tuple

```
INSERT INTO CONTO VALUES('0001', 101, '4916620184642536',NULL);

INSERT INTO CONTO VALUES('0002', 102, '4916808663258482',NULL);

INSERT INTO CONTO VALUES('0003', 103, '4485573546155772',NULL);

INSERT INTO CONTO VALUES('0004', 104, '6011450050067868',NULL);

INSERT INTO CONTO VALUES('0005', 105, '6011624416498112',NULL);

INSERT INTO CONTO VALUES('0006', 106, '6011204046303272',NULL);

INSERT INTO CONTO VALUES('0007', 107, '3045011056435083',NULL);

INSERT INTO CONTO VALUES('0008', 108, '3016599551103201',NULL);

INSERT INTO CONTO VALUES('0009', 109, '3011512846189401',NULL);

INSERT INTO CONTO VALUES('0010', 110, '4175002322463297',NULL);

INSERT INTO CONTO VALUES('0011', 111, '4917571538297247',NULL);

INSERT INTO CONTO VALUES('0012', 112, '3761280974303211',NULL);

INSERT INTO CONTO VALUES('0013', 113, '4917166130059358',NULL);

INSERT INTO CONTO VALUES('0014', 114, '5163448460251351',NULL);

INSERT INTO CONTO VALUES('0015', 115, '5313473422589580',NULL);

INSERT INTO CONTO VALUES('0016', 116, '2221002749460106',NULL);

INSERT INTO CONTO VALUES('0017', 117, '3537118482469013',NULL);

INSERT INTO CONTO VALUES('0018', 118, '3542161225435386',NULL);

INSERT INTO CONTO VALUES('0019', 119, '5163448460251351',NULL);

INSERT INTO CONTO VALUES('0020', 120, '4485573546155772',NULL);

INSERT INTO CONTO VALUES('0021', 121, '4175002322463297',NULL);

INSERT INTO CONTO VALUES('0022', 122, '4916808663258482',NULL);

INSERT INTO CONTO VALUES('0023', 123, '3542161225435386',NULL);

INSERT INTO CONTO VALUES('0024', 124, '3011512846189401',NULL);

INSERT INTO CONTO VALUES('0025', 125, '5313473422589580',NULL);

INSERT INTO CONTO VALUES('0026', 126, '6011450050067868',NULL);

INSERT INTO CONTO VALUES('0027', 127, '4916620184642536',NULL);

INSERT INTO CONTO VALUES('0028', 128, '3537118482469013',NULL);

INSERT INTO CONTO VALUES('0029', 129, '5163448460251351',NULL);
```

```
INSERT INTO CONTO VALUES('0030', 130, '4485573546155772',NULL);
INSERT INTO CONTO VALUES('0031', 131, '5163448460251351',NULL);
INSERT INTO CONTO VALUES('0032', 132, '3011512846189401',NULL);
INSERT INTO CONTO VALUES('0033', 133, '3045011056435083',NULL);
INSERT INTO CONTO VALUES('0034', 134, '2221002749460106',NULL);
```

## Menu

```
INSERT INTO MENU VALUES(1, 'BAMBINO');
INSERT INTO MENU VALUES(2, 'VEGETARIANO');
INSERT INTO MENU VALUES(3, 'VEGANO');
INSERT INTO MENU VALUES(4, 'MARE');
INSERT INTO MENU VALUES(5, 'TERRA');
```

## Selezione

```
INSERT INTO SELEZIONA VALUES( 3, 101, 1);
INSERT INTO SELEZIONA VALUES( 2, 101, 4);
INSERT INTO SELEZIONA VALUES( 4, 102, 1);
INSERT INTO SELEZIONA VALUES( 2, 102, 2);
INSERT INTO SELEZIONA VALUES( 2, 102, 3);
INSERT INTO SELEZIONA VALUES( 2, 102, 5);
INSERT INTO SELEZIONA VALUES( 6, 103, 1);
INSERT INTO SELEZIONA VALUES( 4, 103, 4);
INSERT INTO SELEZIONA VALUES( 5, 103, 5);
INSERT INTO SELEZIONA VALUES( 4, 104, 1);
INSERT INTO SELEZIONA VALUES( 2, 104, 3);
INSERT INTO SELEZIONA VALUES( 2, 104, 4);
INSERT INTO SELEZIONA VALUES( 2, 105, 3);
INSERT INTO SELEZIONA VALUES( 1, 106, 1);
INSERT INTO SELEZIONA VALUES( 3, 106, 5);
INSERT INTO SELEZIONA VALUES( 1, 107, 1);
INSERT INTO SELEZIONA VALUES( 1, 107, 5);
INSERT INTO SELEZIONA VALUES( 3, 108, 1);
INSERT INTO SELEZIONA VALUES( 5, 108, 5);
INSERT INTO SELEZIONA VALUES( 5, 109, 1);
INSERT INTO SELEZIONA VALUES( 3, 109, 3);
INSERT INTO SELEZIONA VALUES( 2, 109, 2);
INSERT INTO SELEZIONA VALUES( 3, 109, 4);
```

INSERT INTO SELEZIONA VALUES( 1, 109, 5);

INSERT INTO SELEZIONA VALUES( 2, 110, 3);

INSERT INTO SELEZIONA VALUES( 5, 110, 4);

INSERT INTO SELEZIONA VALUES( 5, 110, 5);

INSERT INTO SELEZIONA VALUES( 3, 110, 2);

INSERT INTO SELEZIONA VALUES( 3, 111, 5);

INSERT INTO SELEZIONA VALUES( 2, 112, 1);

INSERT INTO SELEZIONA VALUES( 2, 111, 4);

INSERT INTO SELEZIONA VALUES( 3, 113, 1);

INSERT INTO SELEZIONA VALUES( 3, 113, 5);

INSERT INTO SELEZIONA VALUES( 2, 113, 3);

INSERT INTO SELEZIONA VALUES( 2, 114, 2);

INSERT INTO SELEZIONA VALUES( 3, 115, 4);

INSERT INTO SELEZIONA VALUES( 2, 115, 2);

INSERT INTO SELEZIONA VALUES( 3, 116, 4);

INSERT INTO SELEZIONA VALUES( 2, 117, 2);

INSERT INTO SELEZIONA VALUES( 1, 117, 1);

INSERT INTO SELEZIONA VALUES( 5, 118, 5);

INSERT INTO SELEZIONA VALUES( 2, 119, 1);

INSERT INTO SELEZIONA VALUES( 2, 119, 3);

INSERT INTO SELEZIONA VALUES( 1, 120, 1);

INSERT INTO SELEZIONA VALUES( 3, 120, 4);

INSERT INTO SELEZIONA VALUES( 2, 121, 2);

INSERT INTO SELEZIONA VALUES( 2, 122, 4);

INSERT INTO SELEZIONA VALUES( 2, 123, 2);

INSERT INTO SELEZIONA VALUES( 5, 124, 5);

INSERT INTO SELEZIONA VALUES( 2, 124, 3);

INSERT INTO SELEZIONA VALUES( 2, 125, 4);

INSERT INTO SELEZIONA VALUES( 1, 126, 1);

INSERT INTO SELEZIONA VALUES( 6, 126, 4);

INSERT INTO SELEZIONA VALUES( 6, 127, 5);

INSERT INTO SELEZIONA VALUES( 4, 127, 3);

INSERT INTO SELEZIONA VALUES( 2, 128, 3);

INSERT INTO SELEZIONA VALUES( 5, 129, 1);

INSERT INTO SELEZIONA VALUES( 2, 130, 2);

INSERT INTO SELEZIONA VALUES( 4, 131, 5);

INSERT INTO SELEZIONA VALUES( 5, 132, 4);

INSERT INTO SELEZIONA VALUES( 3, 133, 3);

INSERT INTO SELEZIONA VALUES( 2, 133, 2);

INSERT INTO SELEZIONA VALUES( 10, 134, 4);

INSERT INTO SELEZIONA VALUES( 4, 134, 5);

## Pasto

INSERT INTO PASTO VALUES('PENNETTE AL SUGO',8);

INSERT INTO PASTO VALUES('COTOLETTA',10);

INSERT INTO PASTO VALUES('PATATE',2);

INSERT INTO PASTO VALUES('GNOCCHI DI SEMOLINO',10);

INSERT INTO PASTO VALUES('VERDURE RIPIENE',12);

INSERT INTO PASTO VALUES('CESTO DI PANE',3);

INSERT INTO PASTO VALUES('BROCCA DI ACQUA',2);

INSERT INTO PASTO VALUES('BROCCA DI COCA-COLA',3);

INSERT INTO PASTO VALUES('BROCCA DI VINO',6);

INSERT INTO PASTO VALUES('RISOTTO AI PISELLI', 8);

INSERT INTO PASTO VALUES('BURGER VEGETALI', 7);

INSERT INTO PASTO VALUES('CESTO PANE DI SEGALE', 2);

INSERT INTO PASTO VALUES('INSALATA DI MARE', 15);

INSERT INTO PASTO VALUES('LINGUINI AI FRUTTI DI MARE', 18);

INSERT INTO PASTO VALUES('FILETTO D ORATA SCALOPPATO', 18);

INSERT INTO PASTO VALUES('PROSCIUTTO E MOZZARELLA', 9);

INSERT INTO PASTO VALUES('GNOCCHI AL POMODORO', 12);

INSERT INTO PASTO VALUES('FILETTO DI MAIALE', 15);

## Contiene

INSERT INTO CONTIENE VALUES (1,'PENNETTE AL SUGO');

INSERT INTO CONTIENE VALUES (1,'COTOLETTA');

INSERT INTO CONTIENE VALUES (1,'PATATE');

INSERT INTO CONTIENE VALUES (1,'CESTO DI PANE');

INSERT INTO CONTIENE VALUES (1,'BROCCA DI COCA-COLA');

INSERT INTO CONTIENE VALUES (2,'GNOCCHI DI SEMOLINO');

INSERT INTO CONTIENE VALUES (2,'VERDURE RIPIENE');

INSERT INTO CONTIENE VALUES (2,'BROCCA DI ACQUA');

INSERT INTO CONTIENE VALUES (2,'CESTO DI PANE');

INSERT INTO CONTIENE VALUES (3,'RISOTTO AI PISELLI');

INSERT INTO CONTIENE VALUES (3,'BURGER VEGETALI');

INSERT INTO CONTIENE VALUES (3,'CESTO PANE DI SEGALE');

INSERT INTO CONTIENE VALUES (3,'BROCCA DI ACQUA');

INSERT INTO CONTIENE VALUES (4,'LINGUINI AI FRUTTI DI MARE');

INSERT INTO CONTIENE VALUES (4,'FILETTO D ORATA SCALOPPATO');

INSERT INTO CONTIENE VALUES (4,'CESTO DI PANE');

INSERT INTO CONTIENE VALUES (4,'BROCCA DI VINO');

INSERT INTO CONTIENE VALUES (4,'INSALATA DI MARE');

INSERT INTO CONTIENE VALUES (5,'PROSCIUTTO E MOZZARELLA');

INSERT INTO CONTIENE VALUES (5,'GNOCCHI AL POMODORO');

INSERT INTO CONTIENE VALUES (5,'FILETTO DI MAIALE');

INSERT INTO CONTIENE VALUES (5,'CESTO DI PANE');

INSERT INTO CONTIENE VALUES (5,'BROCCA DI VINO');

INSERT INTO CONTIENE VALUES (5,'PATATE');

## **Ingrediente**

INSERT INTO INGREDIENTE VALUES('AA000001','PASTA');

INSERT INTO INGREDIENTE VALUES('AA000002','FARINA');

INSERT INTO INGREDIENTE VALUES('AA000003','UOVA');

INSERT INTO INGREDIENTE VALUES('AA000004','OLIO');

INSERT INTO INGREDIENTE VALUES('AA000005','SALE');

INSERT INTO INGREDIENTE VALUES('AA000006','PEPE');

INSERT INTO INGREDIENTE VALUES('AA000007','RISO');

INSERT INTO INGREDIENTE VALUES('AA000008','SALSA');

INSERT INTO INGREDIENTE VALUES('AA000009','POLLO');

INSERT INTO INGREDIENTE VALUES('AA000010','PATATE');

INSERT INTO INGREDIENTE VALUES('AA000011','SEMOLINO');

INSERT INTO INGREDIENTE VALUES('AA000012','INSALATA');

INSERT INTO INGREDIENTE VALUES('AA000013','PANE');

INSERT INTO INGREDIENTE VALUES('AA000014','ACQUA');

INSERT INTO INGREDIENTE VALUES('AA000015','COCA-COLA');

INSERT INTO INGREDIENTE VALUES('AA000016','VINO');

INSERT INTO INGREDIENTE VALUES('AA000017','PISELLI');

INSERT INTO INGREDIENTE VALUES('AA000018','BURGER VEGETALI');

INSERT INTO INGREDIENTE VALUES('AA000019','PANE DI SEGALE');

INSERT INTO INGREDIENTE VALUES('AA000020','FRUTTI DI MARE');

INSERT INTO INGREDIENTE VALUES('AA000021','ZUCCHINE');

INSERT INTO INGREDIENTE VALUES('AA000022','PEPERONI');

INSERT INTO INGREDIENTE VALUES('AA000023','CIPOLLA');

INSERT INTO INGREDIENTE VALUES('AA000024','RICOTTA');

INSERT INTO INGREDIENTE VALUES('AA000025','PARMIGIANO');

INSERT INTO INGREDIENTE VALUES('AA000026','MELANZANE');

INSERT INTO INGREDIENTE VALUES('AA000027','CARCIOFI');

INSERT INTO INGREDIENTE VALUES('AA000028','AGLIO');

INSERT INTO INGREDIENTE VALUES('AA000029','GAMBERI');

INSERT INTO INGREDIENTE VALUES('AA000030','MERLUZZETTI');

INSERT INTO INGREDIENTE VALUES('AA000031','ALICI');

INSERT INTO INGREDIENTE VALUES('AA000032','CALAMARI');

INSERT INTO INGREDIENTE VALUES('AA000033','MOSCARDINI');

INSERT INTO INGREDIENTE VALUES('AA000034','LIMONE');

INSERT INTO INGREDIENTE VALUES('AA000035','POMODORINI');

INSERT INTO INGREDIENTE VALUES('AA000036','POMODORI');

INSERT INTO INGREDIENTE VALUES('AA000037','PREZZEMOLO');

INSERT INTO INGREDIENTE VALUES('AA000038','PEPERONCINO');

INSERT INTO INGREDIENTE VALUES('AA000039','PROSCIUTTO COTTO');

INSERT INTO INGREDIENTE VALUES('AA000040','PROSCIUTTO CRUDO');

INSERT INTO INGREDIENTE VALUES('AA000041','MOZZARELLA DI BUFALA');

INSERT INTO INGREDIENTE VALUES('AA000042','MAIALE');

INSERT INTO INGREDIENTE VALUES('AA000043','ORATA');

INSERT INTO INGREDIENTE VALUES('AA000044','BASILICO');

INSERT INTO INGREDIENTE VALUES('AA000045','PAN GRATTATO');

## Si compone

INSERT INTO SI\_COMPONE VALUES('PENNETTE AL SUGO', 'AA000001', '250g');

INSERT INTO SI\_COMPONE VALUES('PENNETTE AL SUGO', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('PENNETTE AL SUGO', 'AA000008', '100ml');

INSERT INTO SI\_COMPONE VALUES('PENNETTE AL SUGO', 'AA000005', '5g');

INSERT INTO SI\_COMPONE VALUES('PENNETTE AL SUGO', 'AA000044', '2g');

INSERT INTO SI\_COMPONE VALUES('PENNETTE AL SUGO', 'AA000025', '25g');

INSERT INTO SI\_COMPONE VALUES('COTOLETTA', 'AA000009', '200g');

INSERT INTO SI\_COMPONE VALUES('COTOLETTA', 'AA000005', '2g');

INSERT INTO SI\_COMPONE VALUES('COTOLETTA', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('COTOLETTA', 'AA000045', '125g');

INSERT INTO SI\_COMPONE VALUES('PATATE', 'AA000005', '2g');

INSERT INTO SI\_COMPONE VALUES('PATATE', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('PATATE', 'AA000010', '200g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI DI SEMOLINO', 'AA000001', '250g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI DI SEMOLINO', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI DI SEMOLINO', 'AA000005', '5g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI DI SEMOLINO', 'AA000003', '2');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI DI SEMOLINO', 'AA000011', '150g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000005', '2g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000045', '125g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000003', '2');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000006', '2g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000021', '50g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000022', '50g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000023', '50g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000024', '50g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000026', '50g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000027', '50g');

INSERT INTO SI\_COMPONE VALUES('VERDURE RIPIENE', 'AA000028', '5g');

INSERT INTO SI\_COMPONE VALUES('CESTO DI PANE', 'AA000013', '1kg');

INSERT INTO SI\_COMPONE VALUES('BROCCA DI ACQUA', 'AA000014', '2l');

INSERT INTO SI\_COMPONE VALUES('BROCCA DI COCA-COLA', 'AA000015', '2l');

INSERT INTO SI\_COMPONE VALUES('BROCCA DI VINO', 'AA000016', '2l');

INSERT INTO SI\_COMPONE VALUES('RISOTTO AI PISELLI', 'AA000007', '250g');

INSERT INTO SI\_COMPONE VALUES('RISOTTO AI PISELLI', 'AA000005', '5g');

INSERT INTO SI\_COMPONE VALUES('RISOTTO AI PISELLI', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('RISOTTO AI PISELLI', 'AA000023', '20g');

INSERT INTO SI\_COMPONE VALUES('RISOTTO AI PISELLI', 'AA000017', '200g');

INSERT INTO SI\_COMPONE VALUES('RISOTTO AI PISELLI', 'AA000025', '25g');

INSERT INTO SI\_COMPONE VALUES('BURGER VEGETALI', 'AA000018', '400g');

INSERT INTO SI\_COMPONE VALUES('CESTO PANE DI SEGALE', 'AA000019', '1kg');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000002', '100g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000005', '2g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000006', '2g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000029', '50g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000030', '50g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000031', '50g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000032', '50g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000033', '50g');

INSERT INTO SI\_COMPONE VALUES('INSALATA DI MARE', 'AA000034', '1');

INSERT INTO SI\_COMPONE VALUES('LINGUINI AI FRUTTI DI MARE', 'AA000001', '250g');

INSERT INTO SI\_COMPONE VALUES('LINGUINI AI FRUTTI DI MARE', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('LINGUINI AI FRUTTI DI MARE', 'AA000005', '5g');

INSERT INTO SI\_COMPONE VALUES('LINGUINI AI FRUTTI DI MARE', 'AA000038', '5g');

INSERT INTO SI\_COMPONE VALUES('LINGUINI AI FRUTTI DI MARE', 'AA000028', '2g');

INSERT INTO SI\_COMPONE VALUES('LINGUINI AI FRUTTI DI MARE', 'AA000020', '150g');

INSERT INTO SI\_COMPONE VALUES('FILETTO D ORATA SCALOPPATO', 'AA000028', '2g');

INSERT INTO SI\_COMPONE VALUES('FILETTO D ORATA SCALOPPATO', 'AA000043', '300g');

INSERT INTO SI\_COMPONE VALUES('FILETTO D ORATA SCALOPPATO', 'AA000037', '10g');

INSERT INTO SI\_COMPONE VALUES('FILETTO D ORATA SCALOPPATO', 'AA000005', '2g');



INSERT INTO SI\_COMPONE VALUES('FILETTO D ORATA SCALOPPATO', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('FILETTO D ORATA SCALOPPATO', 'AA000034', '1');

INSERT INTO SI\_COMPONE VALUES('PROSCIUTTO E MOZZARELLA', 'AA000040', '150g');

INSERT INTO SI\_COMPONE VALUES('PROSCIUTTO E MOZZARELLA', 'AA000041', '300g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000001', '250g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000005', '5g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000006', '2g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000008', '100ml');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000028', '2g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000025', '25g');

INSERT INTO SI\_COMPONE VALUES('GNOCCHI AL POMODORO', 'AA000044', '2g');

INSERT INTO SI\_COMPONE VALUES('FILETTO DI MAIALE', 'AA000042', '300g');

INSERT INTO SI\_COMPONE VALUES('FILETTO DI MAIALE', 'AA000004', '10ml');

INSERT INTO SI\_COMPONE VALUES('FILETTO DI MAIALE', 'AA000005', '2g');

INSERT INTO SI\_COMPONE VALUES('FILETTO DI MAIALE', 'AA000006', '2g');

## Fornitore

INSERT INTO FORNITORE VALUES('IT620184642', 'BOLDRINI', '344854646', 'ROMA', '00148', 'ETTORE ROMAGNOLI');

INSERT INTO FORNITORE VALUES('IT916808663', 'AGRUMETTA BEVANDE', '315760081', 'ROMA', '00171', 'GIANDOMENICO ROMAGNOSI');

INSERT INTO FORNITORE VALUES('IT155772217', 'CAPECCHI', '339514323', 'ROMA', '00123', 'PIAZZA VENERE');

INSERT INTO FORNITORE VALUES('IT601162441', 'CENTER FOOD', '340187959', 'ROMA', '00131', 'VIVARO ROMANO');

INSERT INTO FORNITORE VALUES('IT204046303', 'AM CARNI', '338341685', 'ROMA', '00139', 'UMBERTO COROMALDI');

INSERT INTO FORNITORE VALUES('IT001105643', 'COSTIGLIOLA', '339758431', 'NAPOLI', '80100', 'SECONDIGLIANO');

INSERT INTO FORNITORE VALUES('IT512846189', 'CALIENDO BIBITE', '328741675', 'NAPOLI', '80141', 'CAPODICHINO');

INSERT INTO FORNITORE VALUES('IT430321165', 'CARLSBERG', '339855419', 'NAPOLI', '80150', 'VOMERO');

INSERT INTO FORNITORE VALUES('IT430324372', 'ENOSPACE', '332268518', 'NAPOLI', '80152', 'CAPODIMONTE');

INSERT INTO FORNITORE VALUES('IT430353271', 'GESCA', '320235341', 'NAPOLI', '80200', 'ARENACCIA');

INSERT INTO FORNITORE VALUES('IT898989898', 'LA CHIONA', '334898989', 'TORINO', '10121', 'FRANCESCO');

INSERT INTO FORNITORE VALUES('IT171717171', 'PATERA', '335171717', 'TORINO', '10147', 'LA TORRE');

INSERT INTO FORNITORE VALUES('IT232323232', 'CORTILIA', '334232323', 'TORINO', '10150', 'OREFICE');

INSERT INTO FORNITORE VALUES('IT343434343', 'IL FRULLONE', '335343434', 'TORINO', '10134', 'KANT');

INSERT INTO FORNITORE VALUES('IT656565656', 'LUISANA', '334655656', 'TORINO', '10127', 'PITAGORA');

INSERT INTO FORNITORE VALUES('IT778787877', 'IL MARTELLONE', '335787878', 'MILANO', '20147', 'ERACLITO');

INSERT INTO FORNITORE VALUES('IT333311122', 'PARTIAGO', '334333112', 'MILANO', '20150', 'PIAZZA SOCRATE');

INSERT INTO FORNITORE VALUES('IT454545454', 'RETROMI', '335454545', 'MILANO', '20147', 'PIAZZA PLATONE');

INSERT INTO FORNITORE VALUES('IT666676654', 'POSTIGLIA', '334666366', 'MILANO', '20145', 'ARISTOTELE');

INSERT INTO FORNITORE VALUES('IT122221111', 'DELIA', '335122212', 'MILANO', '20156', 'RENE DESCARTES');

## Fornitura

INSERT INTO FORNITURA VALUES(234, 5130, TO\_DATE ('18-MAG-2020','DD -MON - YYYY') , 'IT620184642', 'IT237858945', TO\_DATE ('12-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(312, 3950, TO\_DATE ('22-MAG-2020','DD -MON - YYYY') , 'IT916808663', 'IT237858945', TO\_DATE ('14-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(145, 2685, TO\_DATE ('19-MAG-2020','DD -MON - YYYY') , 'IT155772217', 'IT237858945', TO\_DATE ('12-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(150, 2160, TO\_DATE ('22-MAG-2020','DD -MON - YYYY') , 'IT601162441', 'IT237858945', TO\_DATE ('15-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(201, 1890, TO\_DATE ('21-MAG-2020','DD -MON - YYYY') , 'IT204046303', 'IT237858945', TO\_DATE ('12-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(300, 5340, TO\_DATE ('22-MAG-2020','DD -MON - YYYY') , 'IT001105643', 'IT789415263', TO\_DATE ('15-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(322, 4380, TO\_DATE ('22-MAG-2020','DD -MON - YYYY') , 'IT512846189', 'IT789415263', TO\_DATE ('15-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(108, 2990, TO\_DATE ('19-MAG-2020','DD -MON - YYYY') , 'IT430321165', 'IT789415263', TO\_DATE ('11-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(188, 2610, TO\_DATE ('22-MAG-2020','DD -MON - YYYY') , 'IT430324372', 'IT789415263', TO\_DATE ('15-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(222, 2000, TO\_DATE ('19-MAG-2020','DD -MON - YYYY') , 'IT430353271', 'IT789415263', TO\_DATE ('11-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(170,5150,TO\_DATE ('19-MAG-2020','DD -MON - YYYY'),'IT898989898','IT123456789',TO\_DATE ('12-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(171,4280,TO\_DATE ('20-MAG-2020','DD -MON - YYYY'),'IT171717171','IT123456789',TO\_DATE ('13-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(172,2800,TO\_DATE ('19-MAG-2020','DD -MON - YYYY'),'IT232323232','IT123456789',TO\_DATE ('12-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(173,2755,TO\_DATE ('21-MAG-2020','DD -MON - YYYY'),'IT343434343','IT123456789',TO\_DATE ('13-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(174,1550,TO\_DATE ('21-MAG-2020','DD -MON - YYYY'),'IT656565656','IT123456789',TO\_DATE ('12-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(175,5500,TO\_DATE ('18-MAG-2020','DD -MON - YYYY'),'IT778787877','IT987654321',TO\_DATE ('11-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(176,4120,TO\_DATE ('22-MAG-2020','DD -MON - YYYY'),'IT333311122','IT987654321',TO\_DATE ('14-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(177,2500,TO\_DATE ('18-MAG-2020','DD -MON - YYYY'),'IT454545454','IT987654321',TO\_DATE ('13-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(178,2580,TO\_DATE ('20-MAG-2020','DD -MON - YYYY'),'IT666676654','IT987654321',TO\_DATE ('14-MAG-2020','DD -MON - YYYY'));

INSERT INTO FORNITURA VALUES(179,1700,TO\_DATE ('21-MAG-2020','DD -MON - YYYY'),'IT122211111','IT987654321',TO\_DATE ('15-MAG-2020','DD -MON - YYYY'));

## Proviene da

INSERT INTO PROVIENE\_DA VALUES('AA000001',234,'20KG');

```
INSERT INTO PROVIENE_DA VALUES('AA000002',234,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000003',234,'60PZ');
INSERT INTO PROVIENE_DA VALUES('AA000004',234,'200LT');
INSERT INTO PROVIENE_DA VALUES('AA000005',234,'15KG');
INSERT INTO PROVIENE_DA VALUES('AA000006',234,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000007',234,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000008',234,'80LT');
INSERT INTO PROVIENE_DA VALUES('AA000009',312,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000010',312,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000011',312,'5KG');
INSERT INTO PROVIENE_DA VALUES('AA000012',312,'50KG');
INSERT INTO PROVIENE_DA VALUES('AA000013',312,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000014',312,'10LT');
INSERT INTO PROVIENE_DA VALUES('AA000015',312,'10LT');
INSERT INTO PROVIENE_DA VALUES('AA000016',312,'4LT');
INSERT INTO PROVIENE_DA VALUES('AA000017',312,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000018',312,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000019',145,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000020',145,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000021',145,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000022',145,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000023',145,'4KG');
INSERT INTO PROVIENE_DA VALUES('AA000024',145,'8KG');
INSERT INTO PROVIENE_DA VALUES('AA000025',145,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000026',145,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000027',145,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000028',150,'4KG');
INSERT INTO PROVIENE_DA VALUES('AA000029',150,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000030',150,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000031',150,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000032',150,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000033',150,'40KG');
INSERT INTO PROVIENE_DA VALUES('AA000034',150,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000035',150,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000036',150,'50KG');
INSERT INTO PROVIENE_DA VALUES('AA000037',201,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000038',201,'3KG');
INSERT INTO PROVIENE_DA VALUES('AA000039',201,'40KG');
INSERT INTO PROVIENE_DA VALUES('AA000040',201,'40KG');
```

```
INSERT INTO PROVIENE_DA VALUES('AA000041',201,'50KG');
INSERT INTO PROVIENE_DA VALUES('AA000042',201,'40KG');
INSERT INTO PROVIENE_DA VALUES('AA000043',201,'44KG');
INSERT INTO PROVIENE_DA VALUES('AA000044',201,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000045',201,'20KG');
```

```
INSERT INTO PROVIENE_DA VALUES('AA000001',300,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000002',300,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000003',300,'60PZ');
INSERT INTO PROVIENE_DA VALUES('AA000004',300,'200LT');
INSERT INTO PROVIENE_DA VALUES('AA000005',300,'15KG');
INSERT INTO PROVIENE_DA VALUES('AA000006',300,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000007',300,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000008',300,'80LT');
INSERT INTO PROVIENE_DA VALUES('AA000009',300,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000010',322,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000011',322,'5KG');
INSERT INTO PROVIENE_DA VALUES('AA000012',322,'50KG');
INSERT INTO PROVIENE_DA VALUES('AA000013',322,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000014',322,'10LT');
INSERT INTO PROVIENE_DA VALUES('AA000015',322,'10LT');
INSERT INTO PROVIENE_DA VALUES('AA000016',322,'4LT');
INSERT INTO PROVIENE_DA VALUES('AA000017',322,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000018',322,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000019',108,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000020',108,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000021',108,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000022',108,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000023',108,'4KG');
INSERT INTO PROVIENE_DA VALUES('AA000024',108,'8KG');
INSERT INTO PROVIENE_DA VALUES('AA000025',108,'20KG');
INSERT INTO PROVIENE_DA VALUES('AA000026',108,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000027',108,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000028',108,'4KG');
INSERT INTO PROVIENE_DA VALUES('AA000029',188,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000030',188,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000031',188,'10KG');
INSERT INTO PROVIENE_DA VALUES('AA000032',188,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000033',188,'40KG');
```

```
INSERT INTO PROVIENE_DA VALUES('AA000034',188,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000035',188,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000036',188,'50KG');
INSERT INTO PROVIENE_DA VALUES('AA000037',222,'9KG');
INSERT INTO PROVIENE_DA VALUES('AA000038',222,'3KG');
INSERT INTO PROVIENE_DA VALUES('AA000039',222,'40KG');
INSERT INTO PROVIENE_DA VALUES('AA000040',222,'40KG');
INSERT INTO PROVIENE_DA VALUES('AA000041',222,'50KG');
INSERT INTO PROVIENE_DA VALUES('AA000042',222,'40KG');
INSERT INTO PROVIENE_DA VALUES('AA000043',222,'44KG');
INSERT INTO PROVIENE_DA VALUES('AA000044',222,'30KG');
INSERT INTO PROVIENE_DA VALUES('AA000045',222,'20KG');
```

```
INSERT INTO PROVIENE_DA VALUES ('AA000037',170,'50 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000038',170,'10 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000039',170,'100 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000040',170,'100 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000041',170,'15 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000042',170,'40 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000043',170,'25 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000044',170,'10 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000045',170,'10 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000029',171,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000030',171,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000031',171,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000032',171,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000033',171,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000034',171,'35 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000035',171,'50 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000036',172,'50 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000019',172,'10 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000010',172,'30 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000021',172,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000022',172,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000023',172,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000024',172,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000025',172,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000026',172,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000027',172,'20 KG');
```

INSERT INTO PROVIENE\_DA VALUES ('AA000028',172,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000009',173,'25 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000010',173,'50 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000011',173,'10 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000012',173,'50 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000013',173,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000014',173,'100 LT');  
INSERT INTO PROVIENE\_DA VALUES ('AA000015',173,'100 LT');  
INSERT INTO PROVIENE\_DA VALUES ('AA000016',173,'40 LT');  
INSERT INTO PROVIENE\_DA VALUES ('AA000017',173,'30 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000018',173,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000001',174,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000002',174,'30 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000003',174,'50 PZ');  
INSERT INTO PROVIENE\_DA VALUES ('AA000004',174,'100 LT');  
INSERT INTO PROVIENE\_DA VALUES ('AA000005',174,'20 PZ');  
INSERT INTO PROVIENE\_DA VALUES ('AA000006',174,'20 PZ');  
INSERT INTO PROVIENE\_DA VALUES ('AA000007',174,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000008',174,'50 LT');

INSERT INTO PROVIENE\_DA VALUES ('AA000037',175,'50 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000038',175,'10 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000039',175,'100 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000040',175,'100 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000041',175,'15 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000042',175,'40 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000043',175,'25 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000044',175,'10 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000045',175,'10 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000029',176,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000030',176,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000031',176,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000032',176,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000033',176,'20 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000034',176,'35 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000035',176,'50 KG');  
INSERT INTO PROVIENE\_DA VALUES ('AA000036',176,'50 KG');

```

INSERT INTO PROVIENE_DA VALUES ('AA000019',177,'10 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000010',177,'30 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000021',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000022',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000023',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000024',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000025',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000026',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000027',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000028',177,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000009',178,'25 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000010',178,'50 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000011',178,'10 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000012',178,'50 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000013',178,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000014',178,'100 LT');
INSERT INTO PROVIENE_DA VALUES ('AA000015',178,'100 LT');
INSERT INTO PROVIENE_DA VALUES ('AA000016',178,'40 LT');
INSERT INTO PROVIENE_DA VALUES ('AA000017',178,'30 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000018',178,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000001',179,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000002',179,'30 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000003',179,'50 PZ');
INSERT INTO PROVIENE_DA VALUES ('AA000004',179,'100 LT');
INSERT INTO PROVIENE_DA VALUES ('AA000005',179,'20 PZ');
INSERT INTO PROVIENE_DA VALUES ('AA000006',179,'20 PZ');
INSERT INTO PROVIENE_DA VALUES ('AA000007',179,'20 KG');
INSERT INTO PROVIENE_DA VALUES ('AA000008',179,'50 LT');

```

## Trigger

### No same date

Il seguente trigger controlla se un cliente abbia o meno effettuato una prenotazione per la stessa data. In caso affermativo la prenotazione non può concretizzarsi.

#### Implementazione

```

CREATE OR REPLACE TRIGGER NO_SAME_DATE
BEFORE INSERT OR UPDATE ON PRENOTAZIONE
FOR EACH ROW
SAME_DATE EXCEPTION;

```

```

CONTATORE2 NUMBER(3,0);

BEGIN

--CONTROLLO QUANTE TUPLE ESISTONO NELLA TABELLA PRENOTAZIONE CON LO STESSO ID_CL E STESSA DATA
SELECT COUNT(*) INTO CONTATORE2 FROM PRENOTAZIONE WHERE ID_CL= :NEW.ID_CL AND DATA_P=:NEW.DATA_P ;

--SE IL CONTATORE RITORNA UN VALORE MAGGIORE DI ZERO CIOE ESISTE GIA UNA TUPLA CON LA STESSA DATA ALLORA ATTIVA L' ERRORE

IF CONTATORE2 > 0 THEN

RAISE SAME_DATE;

END IF;

EXCEPTION

WHEN SAME_DATE THEN

RAISE_APPLICATION_ERROR(-20005,'PRENOTAZIONE RIFIUTATA: HAI GIA PRENOTATO');

END;

/

```

## Un ristorante massimo

Tale trigger controlla se si sta cercando di assumere un dipendente che è già in carica presso un altro ristorante della nostra catena.

### Implementazione

```

CREATE TRIGGER MAX_1_RIST

BEFORE INSERT OR UPDATE ON LAVORA

FOR EACH ROW

TROPPI_RISTORANTI EXCEPTION;

CONTATORE NUMBER (3,0);

BEGIN

--OCCORRE CONTARE QUANTE VOLTE APPARE UN DIPENDENTE NELLA TABELLA LAVORA CHE ASSOCIA QUEST'ULTIMO AD UN RISTORANTE

SELECT COUNT(ID_STAFF) INTO CONTATORE FROM LAVORA WHERE ID_STAFF=:NEW.ID_STAFF;

IF CONTATORE >= 1 THEN

RAISE TROPPI_RISTORANTI;

END IF;

EXCEPTION

WHEN TROPPI_RISTORANTI THEN

RAISE_APPLICATION_ERROR(-20004,'Il dipendente lavora gia per un ristorante');

END;

/

```

## Massimo sedici dipendenti

Il trigger che segue verifica se un ristorante che ha già sedici dipendenti sta tentando l'inserimento di un nuovo staff, in caso affermativo il trigger nega tale inserimento.

### Implementazione

```

CREATE OR REPLACE TRIGGER MASSIMO_16_DIP

BEFORE INSERT OR UPDATE ON LAVORA

```



```

FOR EACH ROW

TROPPI_DIPENDENTI EXCEPTION;

CONTATORE NUMBER(2,0);

BEGIN

--CONTA IL NUMERO DI TUPLE IN LAVORA PER IL RISTORANTE CON ID_RIST UGUALE

SELECT COUNT (ID_RIST) INTO CONTATORE FROM LAVORA WHERE ID_RIST=:NEW.ID_RIST;

IF CONTATORE >= 16 THEN

RAISE TROPPI_DIPENDENTI;

END IF;

EXCEPTION

WHEN TROPPI_DIPENDENTI THEN

RAISE_APPLICATION_ERROR(-20003,'Il ristorante supera il numero massimo di dipendenti consentiti.');
```

END;

/

## Controllo tavolo ristorante

Il trigger verifica se il tavolo associato ad una prenotazione è effettivamente presente nel ristorante per il quale l'utente ha prenotato.

### Implementazione

```

CREATE OR REPLACE TRIGGER CHECK_RIST_TAV

BEFORE INSERT OR UPDATE ON ORDINE_SALA

FOR EACH ROW

RISTORANTE_TAVOLO EXCEPTION;

RISTO_PREN VARCHAR2(20);

BEGIN

--SELEZIONIAMO L'ID DEL RISTORANTE ASSOCIATO A QUELL' ORDINE

SELECT ID_RISTORANTE INTO RISTO_PREN FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=(SELECT ID_PRENOTAZIONE FROM ORDINE

WHERE N_ORDINE=:NEW.N_ORDINE);

--CONTROLLA SE L'ID DEL RISTORANTE ASSOCIATO A QUELL'ORDINE SIA DIVERSO DA QUELLO SU ORDINE_SALA ASSOCIATO AL TAVOLO

IF TO_CHAR(RISTO_PREN) != TO_CHAR(:NEW.ID_RISTORANTE) THEN

RAISE RISTORANTE_TAVOLO;

END IF;

EXCEPTION

WHEN RISTORANTE_TAVOLO THEN

RAISE_APPLICATION_ERROR(-20011,'Il tavolo selezionato non appartiene al ristorante prenotato.');
```

END;

/

## Controllo prenotazione

Il trigger che segue nega la possibilità ad un utente di prenotarsi con due settimane di anticipo .

### Implementazione

```

CREATE OR REPLACE TRIGGER CHECK_ANTICIPO
BEFORE INSERT OR UPDATE ON PRENOTAZIONE
FOR EACH ROW
TROPPO_PRESTO EXCEPTION;

ANTICIPO NUMBER (3,0);

BEGIN

SELECT TRUNC(:NEW.DATA_P) - TRUNC (:NEW.DATA_CHIAMATA) INTO ANTICIPO FROM DUAL;

--CONTROLLA SE LA DIFFERENZA TRA LA DATA IN CUI SI VUOLE EFFETTUARE LA PRENOTAZIONE E LA DATA IN CUI SI EFFETTUA LA
PRENOTAZIONE SIA MAGGIORE DI 14 (NUMERO DI GIORNI)

IF (ANTICIPO > 14 ) THEN

RAISE TROPPO_PRESTO;

END IF;

EXCEPTION

WHEN TROPPO_PRESTO THEN

RAISE_APPLICATION_ERROR(-20060,'Impossibile prenotare con piu di 14 giorni di anticipo.');
```

END;

/

## Controllo Journey

Tale trigger nega la possibilità a chiunque abbia un ruolo diverso da “fattorino” di effettuare la consegna.

### Implementazione

```

CREATE OR REPLACE TRIGGER CHECK_FATTORINO
BEFORE INSERT OR UPDATE ON JOURNEY
FOR EACH ROW
RUOLO_STAFF_ERRATO EXCEPTION;

RUOLO_S VARCHAR2(15);

FATTORINO_TMP VARCHAR2(15);

BEGIN

FATTORINO_TMP := 'FATTORINO';

SELECT RUOLO INTO RUOLO_S FROM STAFF WHERE N_CARTELLINO=:NEW.id_staff;

--CONTROLLA SE IL RUOLO DEL DIPENDENTE ASSOCIATO A JOURNEY SIA UGUALE ALLA STRINGA FATTORINO

IF TO_CHAR(RUOLO_S) != TO_CHAR(FATTORINO_TMP) THEN

RAISE RUOLO_STAFF_ERRATO;

END IF;

EXCEPTION

WHEN RUOLO_STAFF_ERRATO THEN

RAISE_APPLICATION_ERROR(-20008,'Ruolo staff errato ');

END;
```

/

## Controllo giorno libero

Il trigger controlla se la data della prenotazione non coincide con il giorno di chiusura del ristorante. Tale attributo è rappresentato con un numero che indica il giorno di chiusura (1:lunedì, 2:martedì ecc.)

### Implementazione

```
CREATE OR REPLACE TRIGGER CHECK_DAY_OFF
BEFORE INSERT OR UPDATE ON PRENOTAZIONE
FOR EACH ROW
data_errata EXCEPTION;
day_of CHAR(1) ;
BEGIN
--SELEZIONE IL GIORNO LIBERO DAL RISTORANTE CON QUELLA PARTITA IVA
SELECT DAY_OFF INTO day_of FROM RISTORANTE WHERE P_IVA=:NEW.ID_RISTORANTE;
--CONFRONTA SE IL GIORNO NELLA DATA È UGUALE AL GIORNO LIBERO DI QUEL RISTORANTE
IF to_char(:NEW.data_p,'D') = day_of THEN
RAISE data_errata;
END IF;
EXCEPTION
WHEN data_errata THEN
RAISE_APPLICATION_ERROR(-20001,'Stai prenotando nel giorno di chiusura del ristorante');
END ;
/
```

## Controllo data carta di credito

Il trigger controlla se la carta di credito inserita all'atto del pagamento non sia scaduta.

### Implementazione

```
CREATE OR REPLACE TRIGGER CHECK_DATA_CC
BEFORE INSERT OR UPDATE ON CARTA_DI_CREDITO
FOR EACH ROW
SCADENZA_ERROR EXCEPTION;
DATA_OGGI DATE;
BEGIN
SELECT SYSDATE INTO DATA_OGGI FROM DUAL;
--SE LA SCADENZA DELLA CARTA DI CREDITO SIA INFERIORE O UGUALE ALLA DATA DI OGGI
IF :NEW.SCADENZA <= DATA_OGGI THEN
RAISE SCADENZA_ERROR;
END IF;
EXCEPTION
WHEN SCADENZA_ERROR THEN
RAISE_APPLICATION_ERROR(-20002,'PAGAMENTO RIFIUTATO:carta di credito scaduta.');
```

```
END;
```

```
/
```

## Controllo capienza tavolo

Il trigger controlla se il tavolo associato ad un ordine è capiente almeno quanto il numero di persone in prenotazione.

### Implementazione

```
CREATE OR REPLACE TRIGGER check_capienza
```

```
BEFORE INSERT OR UPDATE ON ORDINE_SALA
```

```
FOR EACH ROW
```

```
CAPIENZA_ERRATA EXCEPTION;
```

```
CAP NUMBER(2,0);
```

```
NUM_PERS NUMBER(2,0);
```

```
BEGIN
```

```
--PRENDO IL NUMERO DI PERSONE DELLA PRENOTAZIONE CHE HA EFFETTUATO QUELL'ORDINE
```

```
SELECT N_PERSONE INTO NUM_PERS FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=(SELECT ID_PRENOTAZIONE FROM ORDINE WHERE  
N_ORDINE=:NEW.N_ORDINE);
```

```
--PRENDE LA CAPIENZA DEL TAVOLO ASSOCIATO A QUELL'ORDINE
```

```
SELECT CAPIENZA INTO CAP FROM TAVOLO WHERE N_TAVOLO=:NEW.ID_TAVOLO AND ID_RISTORANTE=:NEW.ID_RISTORANTE;
```

```
--CONTROLLA SE IL NUMERO DI PERSONE SIA MAGGIORE DELLA CAPIENZA DEL TAVOLO
```

```
IF NUM_PERS > CAP THEN
```

```
RAISE CAPIENZA_ERRATA;
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN CAPIENZA_ERRATA THEN
```

```
RAISE_APPLICATION_ERROR(-20010,'Il tavolo selezionato non è abbastanza capiente');
```

```
END;
```

## Procedure

### Seleziona menu

Tale procedura, che può essere avviata sia dal cliente che dal cameriere, automatizza l'inserimento di un menu all'interno della tabella "seleziona menù". Tale procedura prende in input tre parametri: la chiave del cliente, l'identificativo del menu che intende scegliere e la quantità di quel tipo di menu.

### Implementazione

```
CREATE OR REPLACE PROCEDURE SELEZIONA_MENU( PK CLIENTE.N_TEL%TYPE, ID_MENU NUMBER , QUANTITA NUMBER )
```

```
IS
```

```
ERRORE_1 EXCEPTION;
```

```
ID_PREN PRENOTAZIONE.ID_PRENOTAZIONE%TYPE;
```

```
ID_ORDINE ORDINE.N_ORDINE%TYPE;
```

```
CONTATORE NUMBER(3,0);
```

```
BEGIN
```

--CONTA IL NUMERO DI VOLTE IN CUI L'ID ORDINE,ASSOCIATO ALLA PRENOTAZIONE DEL CLIENTE, APPARE NELLA TABELLA ORDINE.

```
SELECT COUNT(N_ORDINE) INTO CONTATORE FROM ORDINE JOIN PRENOTAZIONE ON  
PRENOTAZIONE.ID_PRENOTAZIONE=ORDINE.ID_PRENOTAZIONE WHERE ID_CL=PK;
```

--SE IL NUMERO DI VOLTE È 0 ALLORA L'UTENTE NON È PRENOTATO

```
IF (CONTATORE = 0 ) THEN
```

```
RAISE ERRORE_1;
```

```
ELSE
```

-- PRENDE L'ULTIMA PRENOTAZIONE EFFETTUATA DAL CLIENTE (L'ID ORDINE VIENE GENERATO IN MANIERA CRESCENTE PER CUI L'ULTIMO ORDINE COINCIDE CON IL VALORE PIU GRANDE ASSOCIATO A QUELLA PRENOTAZIONE, CHE RICORDIAMO VIENE GENERATO DALLA PROCEDURA PREN ) E LO ASSOCIAMO AI MENU SCELTI

```
SELECT MAX(N_ORDINE) INTO ID_ORDINE FROM ORDINE JOIN PRENOTAZIONE ON  
PRENOTAZIONE.ID_PRENOTAZIONE=ORDINE.ID_PRENOTAZIONE WHERE ID_CL=PK;
```

```
INSERT INTO SELEZIONA VALUES (QUANTITA,ID_ORDINE,ID_MENU);
```

```
COMMIT;
```

```
DBMS_OUTPUT.PUT_LINE('OPERAZIONE ESEGUITA CON SUCCESSO');
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN ERRORE_1 THEN
```

```
RAISE_APPLICATION_ERROR(-20020,'L'utente selezionato non risulta essere prenotato');
```

```
END;
```

```
/
```

## Pren

Tale procedura permette all'utente di prenotarsi indicando in input : il suo numero di telefono, il numero di persone che comporranno il tavolo, la data per la quale si desidera venire, il ristorante a cui si vuole prenotare e un flag numero per indicare se si sta prenotando un pasto d'asporto o meno.

### Implementazione

```
CREATE OR REPLACE PROCEDURE PREN(PK CLIENTE.N_TEL%TYPE, N_PERS PRENOTAZIONE.N_PERSONE%TYPE, DATA_PRENOTAZIONE  
DATE,ID_RIST PRENOTAZIONE.ID_RISTORANTE%TYPE, FLAG NUMBER)
```

```
IS
```

```
CONTATORE NUMBER(2,0);
```

```
CONTATORE2 NUMBER(2,0);
```

```
ID_PR PRENOTAZIONE.ID_PRENOTAZIONE%TYPE;
```

```
DATA_CHIAM DATE;
```

```
FLAG_ERRATO EXCEPTION;
```

```
NUOVO_ORDINE ORDINE.N_ORDINE%TYPE;
```

```
RIST_ERRATO EXCEPTION;
```

```
BEGIN
```

-- CONTROLLO SE L'UTENTE EFFETTIVAMENTE ESISTE

```
SELECT COUNT(N_TEL) INTO CONTATORE FROM CLIENTE WHERE N_TEL=PK;
```

```
IF (CONTATORE=0) THEN
```

```

RAISE NO_DATA_FOUND;

END IF;

--CONTROLLO SE LA PARTITA IVA ESISTE

SELECT COUNT(P_IVA) INTO CONTATORE2 FROM RISTORANTE WHERE P_IVA=ID_RIST;

IF (CONTATORE2=0) THEN

RAISE RIST_ERRATO;

END IF;

SELECT SYSDATE INTO DATA_CHIAM FROM DUAL;

--GENERO UN NUOVO ID_PRENOTAZIONE;

SELECT MAX(ID_PRENOTAZIONE+1) INTO ID_PR FROM PRENOTAZIONE;

INSERT INTO PRENOTAZIONE VALUES(ID_PR,N_PERS,PK,DATA_CHIAM,ID_RIST,DATA_PRENOTAZIONE);

-- CONTROLLO PER VERIFICARE CHE IL VALORE DEL FLAG SIA AMMESSO, PER DETERMINARE

--SE L'UTENTE STA EFFETTUANDO UNA PRENOTAZIONE PER CASA O IN SALA

IF (FLAG > 1 AND FLAG < 0) THEN

RAISE FLAG_ERRATO;

ELSIF (FLAG=1) THEN

COMMIT;

DBMS_OUTPUT.PUT_LINE('PRENOTAZIONE EFFETTUATA CON SUCCESSO CON ID:' || ID_PR );

DBMS_OUTPUT.PUT_LINE('LA ASPETTIAMO IL GIORNO ' || TO_CHAR(DATA_PRENOTAZIONE, 'DD-MON-YYYY HH24:MI '));

ELSE

--GENERO UN NUOVO NUMERO ORDINE

SELECT MAX(N_ORDINE+1) INTO NUOVO_ORDINE FROM ORDINE;

INSERT INTO ORDINE VALUES (NUOVO_ORDINE,ID_PR);

INSERT INTO ORDINE_CASA VALUES (NUOVO_ORDINE);

COMMIT;

DBMS_OUTPUT.PUT_LINE('PRENOTAZIONE EFFETTUATA CON SUCCESSO, RICHIAMARE LA PROCEDURA SELEZIONA_MENU');

END IF;

EXCEPTION

WHEN NO_DATA_FOUND THEN

RAISE_APPLICATION_ERROR(-20021,'L'utente non risulta essere registrato. Eseguire la registrazione');

WHEN FLAG_ERRATO THEN

RAISE_APPLICATION_ERROR(-20022,'FLAG AMMESSI : 1 PER PRONOTARE IN SALA, 0 PER PRENOTARE D ASPORTO ');

WHEN RIST_ERRATO THEN

RAISE_APPLICATION_ERROR(-20023,'Il ristorante selezionato non esiste ');

END;

/

```

## Disdetta

Tale procedura, avviata dall'utente cliente, permette ad esso di disdire la prenotazione a patto che la prenotazione riguardi una consumazione al ristorante. La procedura prende in input la chiave del cliente (numero di telefono) e la data per cui si intende disdire.

### Implementazione

```
CREATE OR REPLACE PROCEDURE DISDETTA(PK CLIENTE.N_TEL%TYPE, DATA_PREN DATE)
IS
ORDINE_DA_CASA EXCEPTION;
CONTATORE NUMBER(2,0);
ID_ORDINE NUMBER(2,0);
ID_PR PRENOTAZIONE.ID_PRENOTAZIONE%TYPE;

BEGIN
--CERCO LA PRENOTAZIONE EFFETTUATA PER QUELLA DATA
SELECT ID_PRENOTAZIONE INTO ID_PR FROM PRENOTAZIONE WHERE ID_CL=PK AND TRUNC(DATA_P)=TRUNC(DATA_PREN);

--CONTROLLO SE L'ORDINE RIGUARDA UN PASTO DA ASPORTO O MENO
SELECT COUNT(ORDINE_CASA.N_ORDINE) INTO CONTATORE FROM ORDINE_CASA JOIN ORDINE ON
ORDINE_CASA.N_ORDINE=ORDINE.N_ORDINE WHERE ORDINE.ID_PRENOTAZIONE=ID_PR;

IF (CONTATORE > 0) THEN
RAISE ORDINE_DA_CASA;
ELSE
DELETE FROM PRENOTAZIONE WHERE ID_CL=PK AND ID_PRENOTAZIONE=ID_PR;

COMMIT;

DBMS_OUTPUT.PUT_LINE('DISDETTA EFFETTUATA CON SUCCESSO ');

END IF;

EXCEPTION

WHEN ORDINE_DA_CASA THEN

RAISE_APPLICATION_ERROR(-20025,'Impossibile disdire questo tipo di ordine.');
```

```
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20026,'Impossibile disdire questo ordine. Prenotazione inesistente');

END;

/
```

## Assegna tavolo

Tale procedura è avviata dal cameriere per assegnare il tavolo che risulta essere capiente almeno quanto il numero di persone del tavolo. La procedura fallisce se un cliente arriva con più di 30 minuti di ritardo.

### Implementazione

```
CREATE OR REPLACE PROCEDURE ASSEGNA_TAVOLO(PK CLIENTE.N_TEL%TYPE, ID_PR PRENOTAZIONE.ID_PRENOTAZIONE%TYPE)
IS
TROPPO_TARDI EXCEPTION;
TROPPO_PRESTO EXCEPTION;
```

```

CONTATORE NUMBER(2,0);

DATA_PRESENZA DATE;

DIFFERENZA NUMBER(6,0);

numero_persone NUMBER(2,0);

NUMERO_TAVOLO NUMBER (2,0);

N_ORDINE_NUOVO ORDINE.N_ORDINE%TYPE;

CAPIENZA_TAVOLO TAVOLO.CAPIENZA%TYPE;

DATA_PREN DATE;

RIST_RISTORANTE.P_IVA%TYPE;

CHECK_PR PRENOTAZIONE.ID_PRENOTAZIONE%TYPE;

VARIABILE NUMBER(3,0);

BEGIN

--CI SERVE SOLO PER VEDERE SE LA QUERY RITORNA UN NO_DATA_FOUND O MENO

SELECT ID_PRENOTAZIONE INTO CHECK_PR FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=ID_PR ;

SELECT DATA_P INTO DATA_PREN FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=ID_PR;

--Orario in cui arriva il cliente

SELECT SYSDATE INTO DATA_PRESENZA FROM DUAL;

--CONTROLLO SE ARRIVA IN RITARDO

SELECT 24*60* (DATA_PREN-DATA_PRESENZA) INTO DIFFERENZA FROM DUAL;

IF (DIFFERENZA < 30 ) THEN

DELETE FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=ID_PR;

COMMIT;

RAISE TROPPO_TARDI;

ELSIF (DIFFERENZA > 30 ) THEN

RAISE TROPPO_PRESTO;

ELSE

--Assegnazione del tavolo

--TROVO IL NUMERO DI PERSONE

SELECT N_PERSONE INTO NUMERO_PERSONE FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=ID_PR;

--TROVO UN TAVOLO CON UNA CAPIENZA SUPERIORI AL NUMERO DI PERSONE.

SELECT MIN(CAPIENZA) INTO CAPIENZA_TAVOLO FROM TAVOLO WHERE CAPIENZA >=NUMERO_PERSONE AND ID_RISTORANTE=(SELECT
ID_RISTORANTE FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=ID_PR);

--PER RIPRENDERE L'ID RISTORANTE

SELECT ID_RISTORANTE INTO RIST FROM PRENOTAZIONE WHERE ID_PRENOTAZIONE=ID_PR;

SELECT MIN(N_TAVOLO) INTO NUMERO_TAVOLO FROM TAVOLO WHERE CAPIENZA=CAPIENZA_TAVOLO;

--IL CLIENTE È EFFETTIVAMENTE PRESENTE IN SALA PER CUI POSSO GENERARE IL SUO ID ORDINE

SELECT COUNT(N_ORDINE) INTO VARIABILE FROM ORDINE;

IF ( VARIABILE > 0) THEN

SELECT MAX(N_ORDINE+1) INTO N_ORDINE_NUOVO FROM ORDINE;

```



```

--SE È IL PRIMO CLIENTE DA UN EVENTUALE RESET DI ORDINI

ELSE

N_ORDINE_NUOVO:=1;

END IF;

INSERT INTO ORDINE VALUES(N_ORDINE_NUOVO,ID_PR);

INSERT INTO ORDINE_SALA VALUES(N_ORDINE_NUOVO,NUMERO_TAVOLO,RIST);

COMMIT;

DBMS_OUTPUT.PUT_LINE('TAVOLO ASSEGNATO CON SUCCESSO');

END IF;

EXCEPTION

WHEN NO_DATA_FOUND THEN

RAISE_APPLICATION_ERROR(-20027,'La prenotazione non esiste');

WHEN TROPPO_TARDI THEN

RAISE_APPLICATION_ERROR(-20028,'Prenotazione disdetta: utente arrivato troppo tardi');

WHEN TROPPO_PRESTO THEN

RAISE_APPLICATION_ERROR(-20029,'utente arrivato troppo presto');

END;

/

```

## Get conto

Tale procedura, eseguita da un utente di tipo cassiere, serve per comunicare al cliente il totale da pagare e ad inserire tale valore all'interno della tabella conto-

### Implementazione

```

CREATE OR REPLACE PROCEDURE GET_CONTO(N_ORD ORDINE.N_ORDINE%TYPE)

IS

CONTO_INESISTENTE EXCEPTION;

ID_CLI VARCHAR2(11);

CONTROLLO_ERRORE NUMBER(2,0);

QUANTITA_SELEZIONA NUMBER (3,0);

I NUMBER (1,0);

CONTATORE NUMBER(3,0);

PREZZO_PARZIALE NUMBER (4,0);

PREZZO_TOTALE NUMBER (4,0);

PK_CONTO VARCHAR2(8);

CONTA NUMBER(3,0);

CONTA2 NUMBER(3,0);

CC VARCHAR2(16);

BEGIN

--CONTROLLO SE ESISTE IL NUMERO D ORDINE PASSATO IN INPUT

SELECT COUNT(N_ORDINE) INTO CONTROLLO_ERRORE FROM ORDINE WHERE N_ORDINE=N_ORD;

```

```

IF (CONTROLLO_ERRORE = 0 ) THEN

RAISE CONTO_INESISTENTE;

END IF;

PREZZO_TOTALE:=0;

--CICLO FOR PER CALCOLARE IL COSTO MENU SELEZIONATI

FOR I IN 1..5

LOOP

SELECT COUNT(ID_MENU) INTO CONTATORE FROM SELEZIONA WHERE N_ORDINE=N_ORD AND ID_MENU=I;

IF (CONTATORE > 0) THEN

--PRENDO LA QUANTITA DELL I-ESIMO MENU SELEZIONATO

SELECT QUANTITA INTO QUANTITA_SELEZIONA FROM SELEZIONA WHERE N_ORDINE=N_ORD AND ID_MENU=I;

--QUERY PER CALCOLARE IL PREZZO DEL MENU I

--SI PRENDE TUTTI I PREZZI DEI PASTI CHE COMPONGONO L 'I-ESIMO MENU E LI SOMMA

SELECT SUM(prezzo) INTO PREZZO_PARZIALE FROM contiene JOIN pasto ON pasto.nome_pasto=contiene.nome_pasto WHERE id_menu=I;

PREZZO_TOTALE:=PREZZO_TOTALE+(PREZZO_PARZIALE*QUANTITA_SELEZIONA);

END IF;

END LOOP;

DBMS_OUTPUT.PUT_LINE('Totale conto selezionato = ' || PREZZO_TOTALE);

--CONTROLLO SE ESISTONO CODICI O È IL PRIMO CONTO GENERATO

SELECT COUNT(CODICE) INTO CONTA FROM CONTO;

IF (CONTA = 0) THEN

PK_CONTO:='0001';

ELSE

SELECT MAX(CODICE)+1 INTO PK_CONTO FROM CONTO;

END IF;

-- CONTROLLIAMO SE DOBBIAMO FARE UN UPDATE(NEL CASO IN CUI CI FOSSE UN ERRORE UMANO NELL'INSERIMENTO) O UN INSERT

SELECT COUNT(ID_ORDINE) INTO CONTA2 FROM CONTO WHERE ID_ORDINE=N_ORD;

--E' UN UPDATE

IF (CONTA2 >0 ) THEN

UPDATE CONTO SET TOTALE=PREZZO_TOTALE WHERE ID_ORDINE=N_ORD;

COMMIT;

DBMS_OUTPUT.PUT_LINE('UPDATE EFFETTUATO CON SUCCESSO');

ELSE

-- RISALGO A PARTIRE DA N_ORDINE ALLA PK DEL MIO CLIENTE PER TROVARE UNA CARTA DI CREDITO PER PAGARE IL CONTO

SELECT ID_CL INTO ID_CLI FROM PRENOTAZIONE JOIN ORDINE ON PRENOTAZIONE.ID_PRENOTAZIONE=ORDINE.ID_PRENOTAZIONE WHERE N_ORDINE=N_ORD;

--PRENDO UNA CC CORRETTA

SELECT n_card INTO CC FROM carta_di_credito WHERE id_cliente=ID_CLI AND ROWNUM=1;

INSERT INTO CONTO VALUES (PK_CONTO,N_ORD,CC,PREZZO_TOTALE);

```

```

COMMIT;

END IF;

DBMS_OUTPUT.PUT_LINE('Pagamento andato a buon fine');

EXCEPTION

WHEN CONTO_INESISTENTE THEN

RAISE_APPLICATION_ERROR(-20030,'L ordine non esiste.');
```

END;

/

## Get fatturato

Procedura avviata dal proprietario per ottenere il fatturato di un ristorante appartenente alla sua catena, indicato in input-

### Implementazione

```

CREATE OR REPLACE PROCEDURE GET_FATTURATO(RIST RISTORANTE.P_IVA%TYPE)

IS

FATTURATO NUMBER(6,0);

BEGIN

--CALCOLO LA SOMMA DEI TOTALI DEI CONTI ASSOCIATI AD UN RISTORANTE DELLA CATENA

SELECT SUM(TOTALE) INTO FATTURATO FROM CONTO JOIN (SELECT * FROM PRENOTAZIONE JOIN ORDINE ON
PRENOTAZIONE.ID_PRENOTAZIONE=ORDINE.ID_PRENOTAZIONE WHERE ID_RISTORANTE=RIST) ON N_ORDINE=ID_ORDINE;

DBMS_OUTPUT.PUT_LINE('Fatturato= ' || FATTURATO || ' EURO');
```

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20050,' RISTORANTE INESISTENTE');

END;

/

## Viste

### Ordini da effettuare

Tale vista è utile all'utente fattorino per ottenere informazioni sull'ordine .Permette di visualizzare la residenza ed il numero di telefono del cliente(per eventuali disguidi) a cui effettuare la consegna dell'ordine.

--VISTA DELLE INFORMAZIONI SUGLI ORDINI

```

CREATE OR REPLACE VIEW info_per_consegna AS

SELECT RISTORANTE.CITTA,CLIENTE.N_TEL,CLIENTE.CAP,CLIENTE.VIA,ORDINE_CASA.N_ORDINE,PRENOTAZIONE.DATA_P AS
DATA_DA_CONSEGNARE FROM (CLIENTE JOIN PRENOTAZIONE ON CLIENTE.N_TEL= PRENOTAZIONE.ID_CL) JOIN ORDINE ON
ORDINE.ID_PRENOTAZIONE= PRENOTAZIONE.ID_PRENOTAZIONE) JOIN ORDINE_CASA ON ORDINE.N_ORDINE = ORDINE_CASA.N_ORDINE JOIN
RISTORANTE ON RISTORANTE.P_IVA= PRENOTAZIONE.ID_RISTORANTE ORDER BY PRENOTAZIONE.ID_RISTORANTE;
```

## Prenotazioni totali

Tale vista è utile all'utente proprietario per ottenere informazioni sulle prenotazioni totali per ogni ristorante, al fine di visualizzare l'andamento generale.

--VISTA PRENOTAZIONI TOTALI

CREATE OR REPLACE VIEW INFO\_PREN AS

```
SELECT RISTORANTE.CITTA,PRENOTAZIONE.ID_RISTORANTE,COUNT(PRENOTAZIONE.ID_PRENOTAZIONE) AS TOT_PREN FROM PRENOTAZIONE
JOIN RISTORANTE ON RISTORANTE.P_IVA=PRENOTAZIONE.ID_RISTORANTE WHERE PRENOTAZIONE.DATA_P BETWEEN '22-MAG-20' AND
(SELECT SYSDATE FROM DUAL) GROUP BY PRENOTAZIONE.ID_RISTORANTE,RISTORANTE.CITTA;
```

## Data Control Language

### Utente

L'utente "customer" (cliente) può eseguire in autonomia le procedure "pren,disdetta,seleziona\_menu", inoltre può solamente inserire, in modo da non poter visualizzare le informazioni sugli altri clienti e le loro carte di credito, nelle tabelle cliente e carta di credito per registrarsi e caricare la carta di credito

GRANT CONNECT,CREATE SESSION TO CUSTOMER;

GRANT EXECUTE ON PREN TO CUSTOMER;

GRANT EXECUTE ON DISDETTA TO CUSTOMER;

GRANT EXECUTE ON SELEZIONA\_MENU TO CUSTOMER;

GRANT INSERT ON CLIENTE TO CUSTOMER;

GRANT INSERT ON CARTA\_DI\_CREDITO TO CUSTOMER;

GRANT SELECT ON MENU TO CUSTOMER;

### Fattorino

L'utente "fattorino" può vedere i dati presenti sulla vista "info\_per\_consegna" per effettuare le consegne, inoltre deve caricare in Journey ed in consegna i dati alla partenza e aggiornare gli stessi al ritorno. Il permesso delete non è stato dato sulle tabelle Journey e consegna per evitare che un fattorino possa cancellare i dati dei viaggi di altri fattorini.

GRANT CONNECT,CREATE SESSION TO FATTORINO;

GRANT SELECT ON INFO\_PER\_CONSEGNA TO FATTORINO;

GRANT INSERT ON JOURNEY TO FATTORINO;

GRANT SELECT ON JOURNEY TO FATTORINO;

GRANT UPDATE ON JOURNEY TO FATTORINO;

GRANT INSERT ON CONSEGNA TO FATTORINO;

GRANT SELECT ON CONSEGNA TO FATTORINO;

GRANT UPDATE ON CONSEGNA TO FATTORINO;

### Cameriere

L'utente "cameriere" può eseguire la procedura "assegna tavolo" per stabilire a quale cliente assegnare un tavolo una volta arrivato in ristorante e la procedura "seleziona menù" per prendere l'ordine del cliente. Esso, inoltre, ha i permessi di visualizzare le prenotazioni dei clienti per verificare, ad esempio, se il numero di persone della prenotazione coincide con il numero di persone effettivamente arrivate nel ristorante ed inoltre gli sono stati concessi tutti i permessi sulle tabelle ordine ed ordine\_casa.

```
GRANT CONNECT,CREATE SESSION TO CAMERIERE;  
  
GRANT EXECUTE ON ASSEGNA_TAVOLO TO CAMERIERE;  
  
GRANT ALL ON SELEZIONA TO CAMERIERE;  
  
GRANT SELECT ON PRENOTAZIONE TO CAMERIERE;  
  
GRANT EXECUTE ON SELEZIONA_MENU TO CAMERIERE;  
  
GRANT ALL ON ORDINE TO CAMERIERE;  
  
GRANT ALL ON ORDINE_SALA TO CAMERIERE;
```

## Cassiere

L'utente "cassiere" può eseguire la procedura "get conto" per calcolare il totale del conto ed inserire le tuple in conto. Gli sono stati dati tutti i permessi sulla tabella carta di credito per, ad esempio, aggiornare o eliminare le carte di credito scadute già presenti nel database

```
GRANT CONNECT,CREATE SESSION TO CASSIERE;  
  
GRANT EXECUTE ON GET_CONTO TO CASSIERE;  
  
GRANT ALL ON CARTA_DI_CREDITO TO CASSIERE;
```

## Chef

L'utente "chef" ha tutti i permessi sulle tabelle si compone e contiene poiché sta a lui stabilire cosa contiene il menu e con cosa sono preparati i pasti al suo interno

```
GRANT ALL ON SI_COMPONE TO CHEF;  
  
GRANT ALL ON CONTIENE TO CHEF;
```

## Proprietario

L'utente proprietario può eseguire le procedure "get fatturato" e "info pren" per ottenere informazioni rispettivamente sul fatturato totale e sulle prenotazioni di ogni ristorante. Il proprietario della catena dei ristoranti ha tutti i permessi sulla tabella "ristorante" poiché è l'unico a decidere se aprire o chiudere un ristorante nella catena. Ha tutti i permessi sulle tabelle "staff" e "lavora" poiché decide se assumere o licenziare un dipendente ed organizzare i turni di lavoro. Inoltre, è lui a richiedere una fornitura per il ristorante ed a caricare i dati degli ingredienti nel database, e decidere di conseguenza i menu preparabili ed i pasti possibili da cucinare.

```
GRANT CONNECT,CREATE SESSION TO PROPRIETARIO;  
  
GRANT EXECUTE ON GET_FATTURATO TO PROPRIETARIO;  
  
GRANT SELECT ON INFO_PREN TO PROPRIETARIO;  
  
GRANT ALL ON RISTORANTE TO PROPRIETARIO;  
  
GRANT ALL ON STAFF TO PROPRIETARIO;  
  
GRANT ALL ON LAVORA TO PROPRIETARIO;  
  
GRANT ALL ON TAVOLO TO PROPRIETARIO;  
  
GRANT ALL ON MENU TO PROPRIETARIO;  
  
GRANT ALL ON PASTO TO PROPRIETARIO;  
  
GRANT ALL ON PROVIENE_DA TO PROPRIETARIO;  
  
GRANT ALL ON FORNITURA TO PROPRIETARIO;  
  
GRANT ALL ON FORNITORE TO PROPRIETARIO;  
  
GRANT ALL ON INGREDIENTE TO PROPRIETARIO;
```