

# Università degli Studi di Padova ~Dipartimento di Matematica

## Progetto Basi di Dati

A.A. 2016/2017

**Damien Ciagola, Matricola N° 1099107 Username: dciagola**

**Sukhjinder Singh, Matricola N° 1102486 Username: ssingh**

## Nome Progetto: **Fuoco&Fiamme**

### 1) Abstract

IL primo FUOCO & FIAMME (F&F) nasce a Padova nel 1994. Cuore e centro del ristorante sono la tradizionale cucina Mediterranea, preservata in ognuna delle ricette presenti nel menù. Il grande successo ottenuto e la grande richiesta hanno reso necessario, al fine di garantire un supporto adeguato e una facile gestione da parte dei titolari, la creazione di un database in cui la direzione potrà seguire più facilmente le prenotazioni dei clienti, creare e tenere traccia degli ordini con i vari fornitori, e gestire stipendi e mansioni dei propri dipendenti.

### 2) Analisi dei requisiti

Il progetto in esame vuole modellare una base di dati che amministri alcuni aspetti della gestione di un ristorante.

Le entità principali interessate sono: le **Prenotazioni** e gli **Ordini** del locale verso i propri fornitori.

Per ogni **Prenotazione** (identificata attraverso un ID) effettuata da un cliente ci interessa: cognome, n.posti, data e ora. Ogni prenotazione può potenzialmente interessare più **Tavoli**, essi sono identificati dal numero e dalla sala in cui sono situati; di ogni tavolo si vuole memorizzare il numero di posti.

Si vuole tenere traccia di tutte le **Ricevute** di pagamento alla fine di ogni occupazione al tavolo, ogni ricevuta verrà identificata da un codice ed avrà i seguenti attributi: tipo pagamento, data, ora e totale.

Del **personale** si vogliono memorizzare le seguenti informazioni: CF, Nome, Cognome, Sesso, Data di nascita e Data assunzione e Termine contratto, (CF identifica univocamente il dipendente).

Ogni dipendente appartiene a una **Categoria**.

Ogni **Sala** avrà un Codice identificativo, una descrizione e un responsabile che farà parte del personale. solo alcuni dipendenti facenti parte di specifiche categorie (Direttore o Chef) potranno effettuare degli ordini per rifornire il locale.

Gli **Ordini** sono identificati da un numero, ha una data che ne descrive il giorno in cui è stato emesso e lo stato relativo (in corso o concluso). Si vuole sapere chi del personale ha effettuato tale ordine, composto generalmente da più articoli.

Degli **Articoli** si vuole tenere traccia del nome, descrizione ID(chiave) e quantità.

Un ordine è ricevuto da un solo **Fornitore**, ma la ditta opera con più fornitori e ne tiene traccia sul proprio database. Essi hanno una P.Iva, Nome, indirizzo, telefono e una email.

### 3) Glossario dei Termini

<b>Termine</b>	<b>Descrizione</b>	<b>Sinonimi</b>	<b>Collegamenti</b>
Prenotazione	Prenotazione effettuata nel locale da un cliente	\N	Tavoli
Tavoli	Tavolo di una sala del ristorante	\N	Sala, Ricevuta
Ricevute	Ricevuta non fiscale del pagamento relativo ad un tavolo	\N	Tavolo
Personale	Personale/Staff che lavora all'interno del ristorante	Dipendente	Categoria, Sala, Ordini
Categoria	Insieme di mansioni svolte dal personale	Mansione	Personale
Sala	Aree del ristorante contenenti tavoli per il ristoro	\N	Personale,Tavoli
Ordini	Ordini di merci effettuati per conto del ristorante verso un proprio fornitore	\N	Articoli, Personale, Fornitore
Articoli	Merce presente nel ristorante	\N	Ordine
Fornitore	Ditta/impresa che rifornisce il ristorante	\N	Ordine

## 4) Progettazione concettuale

### 4.1) Lista delle classi

- **Prenotazione:**
  - ID INT
  - Data DATE
  - Ora TIME
  - N\_Posti SMALLINT
  - Cognome VARCHAR(20)
- **Tavolo**
  - Numero SMALLINT
  - Posti TINYINT
- **Ricevuta**
  - Codice INT
  - Tipo Pagamento ENUM('Carta di Credito','Contanti','Prepagata')
  - Ora TIME
  - Data DATE
  - Totale DOUBLE
- **Sala**
  - Codice VARCHAR(1)
  - Descrizione VARCHAR(250)
- **Personale**
  - CF VARCHAR(16)
  - Nome VARCHAR(25)
  - Cognome VARCHAR(25)
  - Sesso ENUM('F','M')
  - Data\_Assunzione DATE
  - Termine\_Contratto DATE
  - Data\_Nascita DATE
- **Categoria**
  - Nome VARCHAR(20)
  - Stipendio\_Base DOUBLE
- **Articoli**
  - ID INT
  - Nome VARCHAR(25)
  - Descrizione VARCHAR(250)
  - Quantita' INT

- **Ordine**

-N_Ordine	INT
-Stato	ENUM('In Corso','Completato','Annullato')
-Data	DATE

- **Fornitore**

-P_IVA	VARCHAR(20)
-Nome	VARCHAR(25)
-Indirizzo	VARCHAR(250)
-Telefono	VARCHAR(40)
-E-mail	VARCHAR(50)

## 4.2) Lista delle Associazioni

### **Prenotazioni-Tavolo:**Occupazione

Una prenotazione se supera il numero dei posti del più capiente tavolo a nostra disposizione o, i tavoli utili sono occupati, allora la nostra prenotazione occuperà più di un tavolo (1,N).

Viceversa un tavolo può non essere mai stato occupato, o occupato N volte (0,N).

### **Ricevuta-Tavolo:**Pagamento

Una ricevuta è associata al pagamento di uno ed un solo Tavolo (1,1), e le ricevute relative al pagamento per un tavolo possono essere nulle, o tante quante volte è stato occupato quel tavolo (0,N).

### **Tavolo-Sala:**Ubicazione

Un tavolo è situato in una ed una sola sala (1,1), e, a sua volta una sala contiene almeno un tavolo o N tavoli, (1,N).

### **Sala-Personale:**Responsabile

Un Dipendente (sempre se la sua categoria glielo permette) può Prestare il ruolo di Responsabile (maître) in un'unica sala (0,1), e una sala Deve avere un suo responsabile. (1,1)

### **Personale-Categoria:**Appartenenza

Ogni Dipendente appartiene un'unica categoria, nel caso in cui un cuoco o un cameriere vengono promossi rispettivamente a Chef o maître, abbandonano la precedente mansione e vanno ad occupare la nuova figura(1,1). una Categoria a sua volta può essere completamente vacante o essere occupato da più dipendenti (0,N).

### **Personale-Effettua:**Ordine

All'entità Tavolo, identificata esternamente dall'entità Sala, si è deciso di aggiungere un nuovo attributo Sala identificandolo internamente tramite Sala-Codice.

-Nome                      VARCHAR(20)

- **Articoli**

-ID                              INT

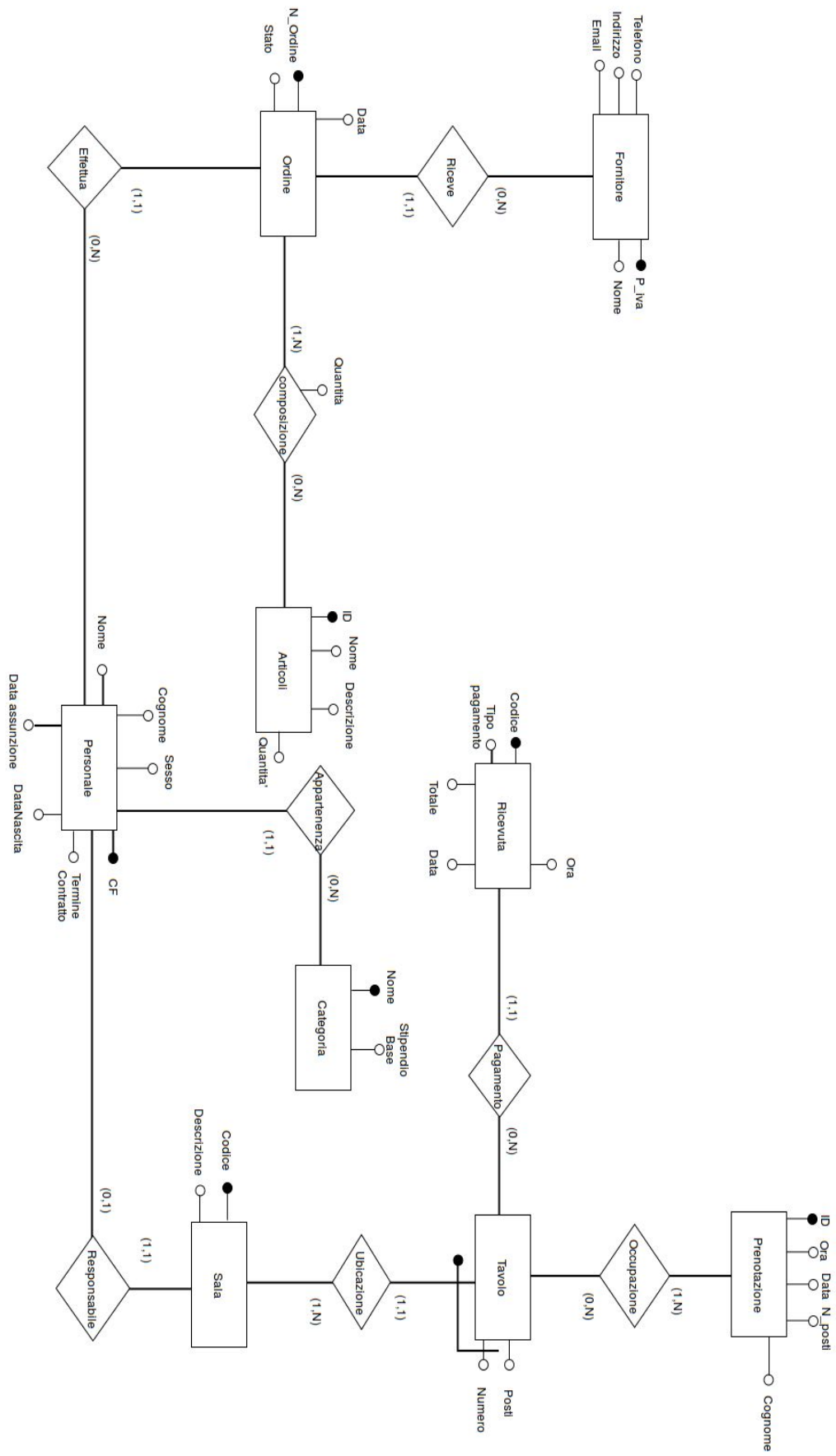
- **Ordine**

-N\_Ordine                      INT

- **Fornitore**

-P\_IVA                              VARCHAR(20)

#### 4.3) SCHEMA E-R



#### **4.4)Vincoli non esprimibili in ER**

4.4.1- Solo uno chef o il direttore possono effettuare un ordine.

4.4.2- Solo il maitre può essere responsabile di una sala.

### **5)Progettazione Logica**

#### **5.1) Schema relazionale**

Sale(Codice, Descrizione, Responsabile\*)

Tavoli(Numero, Sala\*, Posti)

Ricevute(Codice, Tipo Pagamento, Ora, Data, Totale, Tavolo\*, Sala\*)

Prenotazioni(ID, Data, Ora, N\_Posti, Cognome)

Occupazioni(Tavolo\*, Sala\*, Prenotazione\*)

Personale(CF, Nome, Cognome, Sesso, Data\_Nascita, Data\_Assunzione, Termine\_Contratto, Categoria\*)

Categoria(Nome, Stipendio\_Base)

Fornitori(P\_IVA, Nome, Indirizzo, Telefono, E-mail)

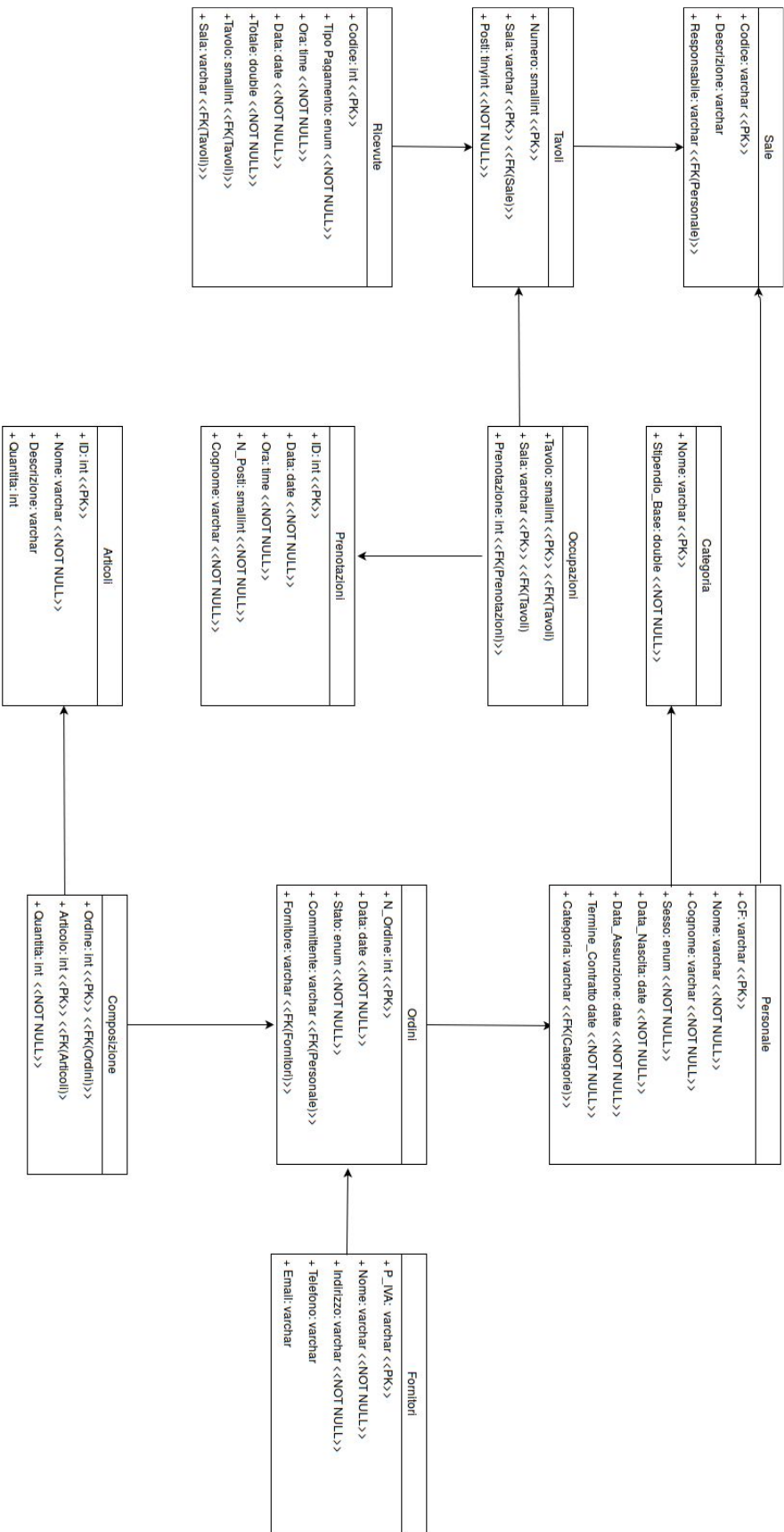
Ordini(N\_Ordine, Data, Stato, Committente\*, Fornitore\*)

Articoli(ID, Nome, Descrizione, Quantita)

Composizione(Ordine\*, Articolo\*, Quantità)

#### **5.2)Schema logico**





## 6)Procedure

- 1. Se la prenotazione è di un numero che non supera il tavolo più grande a disposizione, assegna un tavolo idoneo alla prenotazione.**

```
DELIMITER //
CREATE PROCEDURE AssegnaTavoli(NPosti SMALLINT,Data DATE ,ID INT)
BEGIN
DECLARE nmax SMALLINT;
DECLARE ntav SMALLINT;
DECLARE nsal VARCHAR(1);
SELECT MAX(Tavoli.Posti)
FROM Tavoli
WHERE Tavoli.Numero not in(
                                SELECT Tavolo from Occupazioni, Prenotazioni
                                WHERE Occupazioni.Prenotazione=Prenotazioni.ID AND
                                Prenotazioni.Data=Data
                                )

INTO nmax;
IF NPosti <= nmax THEN
SELECT Tavoli.Numero
FROM Tavoli
WHERE Tavoli.Numero not in(
                                SELECT Occupazioni.Tavolo from Occupazioni, Prenotazioni
                                WHERE Prenotazioni.Data=Data AND
                                Prenotazioni.N_Posti>=NPosti
                                ) LIMIT 1

INTO ntav;

SELECT Tavoli.Sala
FROM Tavoli
WHERE Tavoli.Numero=ntav AND Tavoli.Posti>=NPosti LIMIT 1
INTO nsal;
INSERT INTO Occupazioni (Tavolo,Sala,Prenotazione) VALUES (ntav,nsal,ID);
END IF;
END //
DELIMITER ;
```

- 2. terminata una prenotazione, la elimina.**

```
DELIMITER //
CREATE PROCEDURE Elimina_PRE(PR INT)
BEGIN
DELETE FROM Prenotazioni
WHERE Prenotazioni.ID=PR;
END //
DELIMITER ;
```

## **7)funzioni**

**1. Ritorna il totale degli introiti dell'anno passato per valore.**

```
DELIMITER //
CREATE FUNCTION Introito (ANNO INT)
RETURNS DOUBLE
BEGIN
DECLARE N DOUBLE;
SELECT SUM(Totale) FROM Ricevute R WHERE ANNO=YEAR(R.Data) INTO N;
return N;
END //
DELIMITER ;
```

**2. Dato alla funzione un articolo X, ritorna il numero di volte che è stato ordinato.**

```
DELIMITER //
CREATE FUNCTION NumOrdine(X INT)
RETURNS INT
BEGIN
DECLARE N INT;
SELECT count(*)
FROM Composizione
WHERE Articolo=X INTO N;
return N;
END //
DELIMITER ;
```

**8) Trigger**

**1. Dopo l'inserimento nella tabella Prenotazioni chiama la procedura 6.1**

```
DELIMITER //
CREATE TRIGGER Disposizione
AFTER INSERT ON Prenotazioni
FOR EACH ROW
BEGIN
call AssegnaTavoli(new.N_Posti,new.Data,new.ID);
END //
DELIMITER ;
```

**2. Solo uno chef o un direttore possono effettuare ordini.**

```
DELIMITER //
CREATE TRIGGER Ordinazione
BEFORE INSERT ON Ordini
FOR EACH ROW
BEGIN
IF new.Committente NOT IN (
                        Select CF FROM Personale
                        WHERE Categoria='Direttore' OR Categoria='Chef'
) THEN

DELETE FROM Ordini
WHERE Ordini.Committente=new.Committente;
END IF;
END //
DELIMITER ;
```

**3. Elimina occupazioni e prenotazioni Che hanno Saldato il conto.**

```
DELIMITER //
CREATE TRIGGER EliminaOccupazione
AFTER INSERT ON Ricevute
FOR EACH ROW
BEGIN
DECLARE PR INT;
SELECT Occupazioni.Prenotazione
FROM Occupazioni
WHERE Occupazioni.Tavolo=new.Tavolo AND Occupazioni.Sala=new.Sala INTO PR;
call Elimina_PRE(PR);
END //
DELIMITER ;
```

**9) Query**

### 9.1)

**N\_Ordine, Data e lo Stato di tutti gli ordini ancora in corso, ordinati in modo non decrescente per la Data, con il Nome e Cognome dei committenti e del fornitore.**

```
SELECT N_Ordine, Data, Stato, P.Nome, P.Cognome, F.Nome
FROM Ordini, Personale P, Fornitori F
WHERE Committente=P.CF AND F.P_IVA=Fornitore AND Stato='In Corso' order by Data;
```

N_Ordine	Data	Stato	Nome	Cognome	Nome
11	2017-02-02	In Corso	Maccio	Capatonda	Frigomeccanica s.r.l
12	2017-02-17	In Corso	Rupert	Sciamenna	Forniture Ristoranti
15	2017-02-21	In Corso	Rupert	Sciamenna	Frigomeccanica s.r.l
14	2017-03-03	In Corso	Maccio	Capatonda	Fas Itaila s.r.l

### 9.2)

**Codice, descrizione, numero totale dei tavoli e totale dei posti a sedere per ciascuna sala con il rispettivo responsabile.**

```
SELECT S.Codice, S.Descrizione, COUNT(T.Numero) AS N_Tavoli, SUM(T.Posti) AS
N_Posti, P.Nome AS Responsabile
FROM Tavoli T, Sale S, Personale P
WHERE T.Sala=S.Codice AND P.CF=S.Responsabile
GROUP BY S.Codice;
```

Codice	Descrizione	N_Tavoli	N_Posti	Responsabile
A	Sala lato EST coperta	16	68	Ennio
B	Sala lato OVEST coperta	20	82	Claudia
C	Sala esterna entrata	22	90	Federico

### 9.3)

**Per ciascun fornitore, il nome e quantità dell'articolo maggiormente ordinato.**

```
CREATE VIEW VISTA1(Nome_Fornitore, Nome_Articolo, Quantita) AS
SELECT f.Nome, a.Nome, SUM(c.Quantita)
FROM Ordini o JOIN Fornitori f ON (o.Fornitore=f.P_IVA)
      JOIN Composizione c ON (o.N_Ordine=c.Ordine)
      JOIN Articoli a ON (c.Articolo=a.ID)
GROUP BY o.Fornitore, a.Nome;
```

```
SELECT Nome_Fornitore, Nome_Articolo, Quantita From VISTA1
WHERE Quantita=(
```

```

SELECT MAX(Quantita)
FROM VISTA1 v1
WHERE VISTA1.Nome_Fornitore=v1.Nome_Fornitore
);

```

Nome_Fornitore	Nome_Articolo	Quantita
Nuova Bever-li	Microonde	12
Forniture Ristoranti	Tovaglia Runner	10
NEW Global Vending	Ananas	20
Bragard Italia s.r.l	Frigorifero	7
Frigomeccanica s.r.l	Pasta senza glutine	6
EarArte s.r.l	Carne	2
EarArte s.r.l	Cuocipiada	2
Espress di Rossi	Sfogliatrice	2
Fas Itaila s.r.l	Tostiera	5
For Bar s.r.l	Sfogliatrice	3

#### 9.4)

**per il giorno 18/02/2017 dalle 20 in poi mostrare per ogni sala: descrizione, il numero dei tavoli e posti occupati e il numero dei tavoli e posti ancora disponibili.**

```

Create VIEW A(Descrizione, N_Tavoli_Occupati, N_Posti_Occupati) AS
select S.Descrizione,count(O.Tavolo) AS N_Tavoli_Occupati, SUM(P.N_Posti) AS
N_Posti_Occupati
from Occupazioni O,Prenotazioni P,Sale S
where O.Prenotazione=P.ID AND
P.Data='2017/02/18' AND P.Ora>20 AND
S.Codice=O.Sala
Group By S.Descrizione;

```

```

Create VIEW B(Descrizione, N_Tavoli_Liberi, N_Posti_Liberi) AS
select S.Descrizione,count(T.Numero),SUM(T.Posti)
from Tavoli T, Sale S
where S.Codice=T.Sala
group By S.Descrizione;

```

```

select A.Descrizione, N_Tavoli_Occupati, N_Posti_Occupati,
N_Tavoli_Liberi - N_Tavoli_Occupati AS N_Tavoli_Liberi, N_Posti_Liberi -
N_Posti_Occupati AS N_Posti_Liberi
from A,B where A.Descrizione=B.Descrizione;

```

Descrizione	N_Tavoli_Occupati	N_Posti_Occupati	N_Tavoli_Liberi	N_Posti_Liberi
Sala esterna entrata	4	14	18	76
Sala lato EST coperta	7	21	9	47
Sala lato OVEST coperta	5	24	15	58

9.5)

per ogni sala il tavolo con più incassi in anno 2016

```
select S.Codice,S.Descrizione,R.Tavolo,max(R.Totale) AS Incasso
from Ricevute R,Sale S
where R.Sala=S.Codice AND YEAR(R.Data)=2016
group by S.Codice;
```

Codice	Descrizione	Tavolo	Incasso
A	Sala lato EST coperta	1	178.91
B	Sala lato OVEST coperta	2	181.77
C	Sala esterna entrata	1	103.37

9.6)

nome e cognome degli impiegati che hanno effettuato almeno un numero X di ordini, in questo caso 8.

```
CREATE VIEW N_o_eff(CF,N_Ordini) AS
Select O.Committente,count(*)
from Ordini O
group by O.Committente;
```

```
select P.Nome, P.Cognome
from Personale P, N_o_eff N
where P.CF=N.CF AND N.N_Ordini >=8;
```

Nome	Cognome
Rupert	Sciamenna