**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**

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| 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 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

Un Dipendente (sempre se la sua categoria glielo permette) può disporre dei permessi necessari per effettuare più ordini all’interno del locale(0,N), e un singolo ordine viene effettuato da una sola persona che se ne assume la piena responsabilità(1,1).

**Ordine-Articoli**:Composizione

Un Singolo ordine è composto da almeno uno o più articoli (1,N) e vari articoli possono essere ordinati o meno (0,N).

**Ordine-Fornitore**:Riceve

Un ordine è ricevuto da un solo fornitore (1,1), e esistono fornitori con cui non si hanno mai avuti rapporti (nuovi) e fornitori con cui si hanno regolarmente rapporti di compravendita (0,N).

**4.2.3)Scelta delle chiavi:**

* **Prenotazione:**

-ID INT

* **Tavolo**

-Numero SMALLINT

-Posti TINYINT

=> **4.5)Ristrutturazione**

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

data:image/png;base64,

* **Tavolo**

-Numero SMALLINT FK Sala(Codice VARCHAR(1))

* **Ricevuta**

-Codice INT

* **Sala**

-Codice VARCHAR(1)

* **Personale**

-CF VARCHAR(16)

* **Categoria**

-Nome VARCHAR(20)

* **Articoli**

-ID INT

* **Ordine**

-N\_Ordine INT

* **Fornitore**

-P\_IVA VARCHAR(20)

**4.3) SCHEMA E-R**

Schema_E-R.png

**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**

Schema Logico.png

**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**

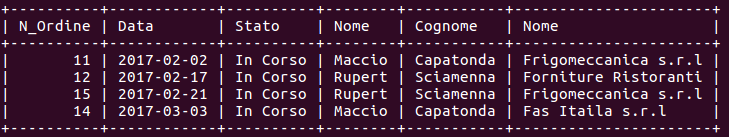
**9.1)**

**N\_Ordine,Data e lo Stato di tutti gli ordini ancora in corso, ordinati in modo non descrescente 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;



**9.2)**

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

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;



**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

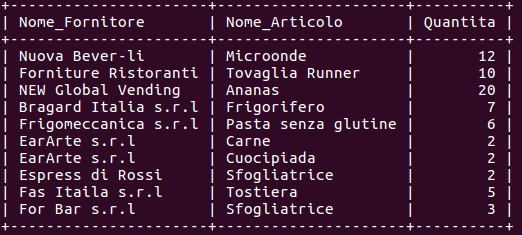
WHERE Quantita=(

SELECT MAX(Quantita)

FROM VISTA1 v1

WHERE VISTA1.Nome\_Fornitore=v1.Nome\_Fornitore

);



**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

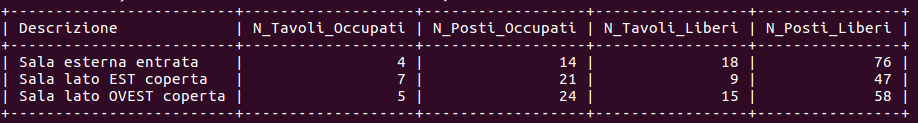
where S.Codice=T.Sala

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;



**9.5)**

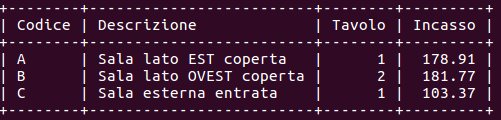
**per ogni sala il tavolo con più incassi in anno 2016**

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;



**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;

