

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

2017/2018

EzTicket

PROGETTO DI BASI DI DATI PER L'UNIVERSITÀ DI PADOVA

GIANMARCO PETTINATO (1068299) & GIUSEPPE VITO BITETTI (1143329)

EzTicket

EzTicket è un software che si propone di agevolare la gestione di fiere, relativi eventi e biglietti seguendo diverse linee guida.

Gli eventi si possono dividere in più giorni ed essere distribuiti su più padiglioni.

La gestione dei biglietti è facilitata tramite la scelta di varie tipologie di tariffe, spaziando da quella a ingresso singolo fino all'abbonamento che perdura per tutta la fiera. Inoltre il software tiene traccia, nel tempo, dei clienti che hanno effettuato un acquisto in modo da agevolare quelli più fedeli.

EzTicket tiene traccia anche di dove si sono svolti i vari eventi e quanti padiglioni sono stati utilizzati per quella specifica fiera. Il programma mette a disposizione varie funzionalità, tra esse l'emissione e la spendibilità del biglietto, con eventuale controllo dei doppietti e/o falsi.

Analisi dei Requisiti

Abbiamo strutturato il database identificando i biglietti relativi agli eventi e alle fiere, con un codice identificativo univoco del biglietto che ne identifica la fiera, l'anno e l'evento, inoltre salvando lo stato del biglietto (Nuovo, Usato, Non più valido).

Ogni biglietto è intestato ad un cliente, identificato da codice fiscale, il quale viene "ricordato" nel sistema.

Le relative fiere vengono diversificate tramite il nome, l'anno, la città ed un codice progressivo; ogni fiera è associata all'eventuale luogo in cui essa si svolge e viene memorizzato dal sistema tenendo traccia dei padiglioni utilizzati e dalla capienza massima totale di essi. Nel caso si tenti di superare la capienza massima di un padiglione il sistema darà errore.

I padiglioni sono identificati attraverso il luogo ed un sistema di numerazione, inoltre viene memorizzata la capienza massima degli stessi per evitare una sovra affluenza.

Gli eventi, oltre che dal codice id, sono caratterizzati dal nome dello stesso, dal padiglione e dalla finestra temporale nella quale si svolgono.

Ciascuna fiera è identificata dalle seguenti informazioni:

- Codice ID che identificherà la fiera in modo univoco;
- Città, dove si svolgerà la fiera;
- Luogo, ovvero la struttura specifica all'interno della città dove si terrà la fiera;
- Anno di svolgimento;
- Data di inizio di tale manifestazione;
- Data di fine di tale manifestazione.

Per ogni padiglione si vuole conoscere:

- Combinazione di lettere e numeri atte ad identificare il padiglione all'interno del luogo e della città;
- Città di ubicazione;
- Luogo, ovvero la struttura dove si trova il padiglione;
- Capienza.

Ogni fiera è composta da almeno un evento il quale avrà le seguenti proprietà:

- Id dell'evento relativo alla fiera;
- Id della fiera di riferimento;
- Eventuale padiglione utilizzato (Potrebbe non utilizzare padiglioni se l'evento è al di fuori di una struttura);
- Giorno e ora di inizio;
- Giorno e ora di fine.

Per ogni evento si ha almeno una tariffa di riferimento con le seguenti informazioni:

- Id della fiera di riferimento;
- Id dell'evento di riferimento;
- Tipo di tariffa (Intera, Base, Ridotta, Abbonamento, ecc.);
- Prezzo;
- Giorni che tale tariffa permette di accedere alla fiera.

Ogni cliente del database sarà composto da:

- Codice fiscale;
- Nome;
- Cognome;
- Data di nascita;
- Eventuale numero di telefono per possibili comunicazioni.

Ogni biglietto acquistato da un cliente verrà caratterizzato da:

- Numero progressivo per l'evento di quella fiera;
- Id della fiera;
- Id dell'evento;
- Tipo di tariffa che è stata applicata alla creazione del biglietto;
- Prezzo pagato;
- Data e ora di emissione del biglietto;
- Stato del biglietto (Se nel lasso di tempo che viene utilizzato è valido o meno);
- Codice fiscale del cliente.

Progettazione concettuale

Biglietto:

Id : int
IdFiera : string
Evento : string
Tipo : string
DEmissione : date
Prezzo : decimal
Stato del biglietto
Valido : date
Non valido : date
Cliente (C.F.) : string

Tariffa

IdFiera : string
IdEvento : string
Tipo : string
Prezzo : decimal
Giorni : int

Cliente:

C.F. : string
Nome : string
Cognome : string
Data di nascita : date
N. Telefono : string

Fiera:

Id : string

Nome : string

Città : string

Luogo : string

Data inizio : date

Data fine : date

Evento:

Id : string

Nome : string

Padiglione : string

IdFiera : string

Data inizio : date

Data fine : date

Padiglione:

Città : string

Luogo : string

Id : string

Capienza : int

Relazioni

Biglietto-Fiera: Emissione

- ☐ Un biglietto viene emesso per una e una sola fiera

Fiera-Biglietto: Emissione

- ☐ Una fiera può non avere biglietti associati

Evento-Tariffa: Prezzo

- ☐ Un Evento deve avere almeno una tariffa

Tariffa-Evento: Prezzo

- ☐ Una tariffa è associata ad uno ed un solo evento

Tariffa-Biglietto: Applicazione

- ☐ Una tariffa può non avere biglietti associati
- ☐ Una sola tariffa può essere applicata a più biglietti

Biglietto-Tariffa: Applicazione

- ☐ Un biglietto ha una e una sola tariffa

Cliente-Biglietto: Acquisto

- ☐ Un cliente può non aver acquistato un biglietto
- ☐ Un cliente può acquistare più biglietti

Biglietto-Cliente: Acquisto

- ☐ Un biglietto è associato ad un ed un solo cliente

Fiera-Evento: Appartenenza

- ☐ Una fiera ha almeno un evento
- ☐ Una fiera può avere più eventi

Evento-Fiera: Appartenenza

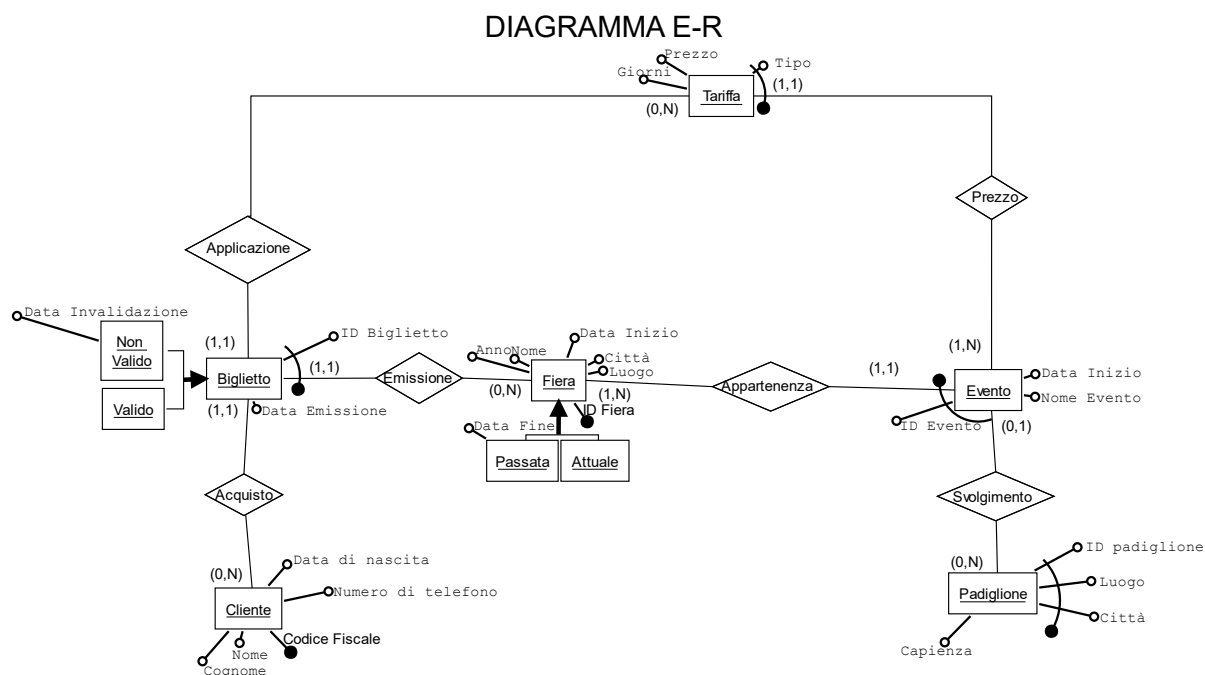
- ❑ Un evento appartiene ad una e una sola fiera

Evento-Padiglione: Svolgimento

- ❑ Un evento può non avere un padiglione
- ❑ Un evento si può svolgere in un ed un solo padiglione

Padiglione-Evento: Svolgimento

- ❑ Un padiglione può essere libero
- ❑ Più padiglioni possono essere occupati da un evento



Ristrutturazione del Diagramma E-R

Per la ristrutturazione del diagramma E-R abbiamo proceduto ad eliminare cicli di dipendenza come quello presente tra Biglietto-Fiera-Evento-Tariffa.

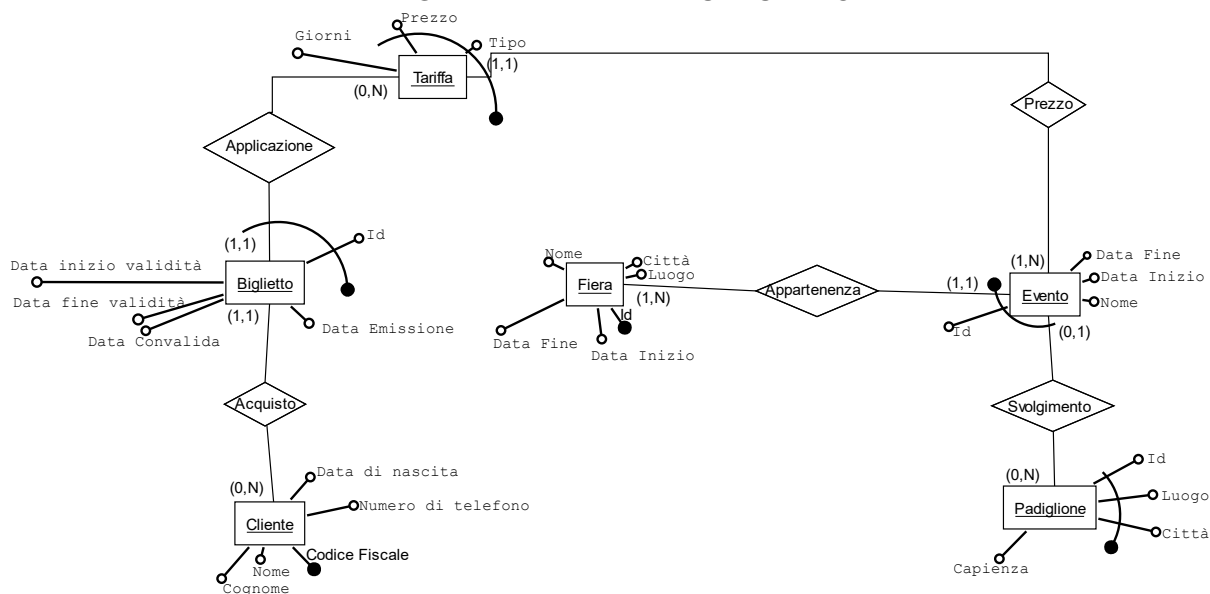
Abbiamo quindi eliminato l'emissione del biglietto da parte di fiera per integrare la dipendenza direttamente dall'entità tariffa.

Abbiamo incorporato nelle entità le generalizzazioni presenti come quelli dell'entità Biglietto e Fiera e quindi aggiunto dei campi a tali entità.

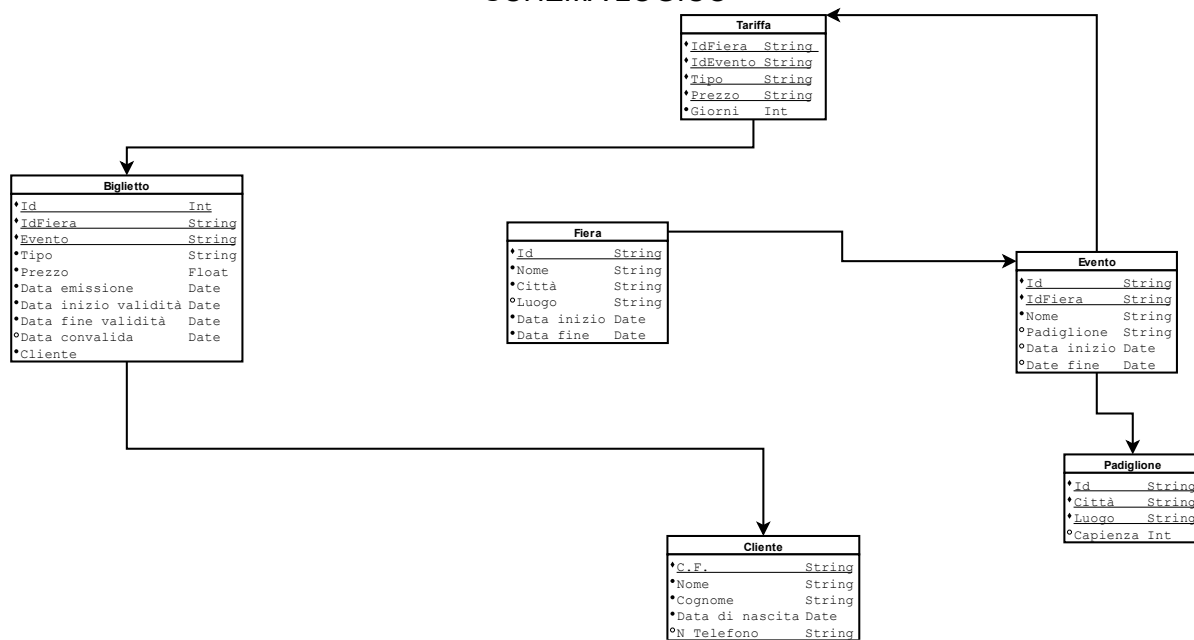
In Biglietto abbiamo inserito Data inizio validità, Data fine validità, Data convalidazione per la gestione della validità del biglietto stesso.

In Fiera abbiamo aggiunto il campo Data fine per distinguere tra le fiere passate e quelle attuali e/o future.

DIAGRAMMA E-R RISTRUTTURATO



SCHEMA LOGICO



Biglietto (Id, IdFiera, Evento, Tipo, Prezzo, Data Emissione, Data inizio validità, Data fine validità, Data convalidazione, Cliente)

Tariffa (IdFiera, IdEvento, Tipo, Prezzo, Giorni)

Cliente (C.F., Nome, Cognome, Data di nascita, N. Telefono)

Fiera (Id , Nome, Città, Luogo, Data inizio, Data fine)

Evento (Nome, Padiglioni, Id, IdFiera, Data inizio, Data fine)

Padiglione (Città, Luogo, Id, Capienza)

QUERY

##Incasso delle Fiere

CREATE VIEW Incasso **AS SELECT** Fiera.Nome, sum(Prezzo) **AS** Incasso **FROM** Biglietto, Fiera **WHERE** Biglietto.IdFiera=Fiera.id **GROUP BY** Biglietto.IdFiera;

Nome	Incasso
Lucca Comix & Games	180.00
Fiera campionaria	101.00
Fiera del Radioamatore ed Elettronica	55.00

Mostra tutte le fiere con i relativi incassi

Fiere con il numero di padiglioni usati

CREATE VIEW NPadUsati **AS SELECT** Fiera.Nome, count(Padiglione) **as** Padiglioni **FROM** Evento **LEFT JOIN** Fiera **ON** Evento.IdFiera=Fiera.Id **group by** Fiera.Id;

Nome	Padiglioni
Lucca Comix & Games	0
Carnevale	0
Carnevale	0
Fiera campionaria	2
Fiera campionaria	1
MantovaComix	1
Fiera del Radioamatore ed Elettronica	2

Mostra tutte le fiere con i padiglioni usati

##Media dei Padiglioni usati in ogni Fiera

CREATE VIEW MediaPadiglioni **AS SELECT** c.Nome, AVG(Padiglioni) **as** Media **FROM** NPadUsati **as** c **group by** c.Nome;

Nome	Media
Carnevale	0.0000
Fiera campionaria	1.5000
Fiera del Radioamatore ed Elettronica	2.0000
Lucca Comix & Games	0.0000
MantovaComix	1.0000

Mostra le fiere con i la media dei padiglioni usati(le fiere non sono separate se hanno lo stesso nome)

##Fiere con i relativi eventi e Tariffario

CREATE VIEW InfoFiere **AS SELECT** Fiera.Nome, Città, Luogo, Padiglione, Evento.Nome **as** NomeEvento, Evento.DataInizio, Evento.DataFine, Tipo, Prezzo **FROM** Fiera **LEFT JOIN** Evento **ON** Fiera.id=Evento.IdFiera **LEFT JOIN** Tariffa **ON** Evento.Id=Tariffa.IdEvento **AND** Tariffa.IdFiera=Evento.IdFiera;

Nome	Città	Luogo	Padiglione	NomeEvento	DataInizio	DataFine	Tipo	Prezzo
Lucca Comix & Games	Lucca	NULL	NULL	Lucca Comix & Games	2018-10-28 00:00:00	2018-10-31 00:00:00	Abb4gg	37.50
Lucca Comix & Games	Lucca	NULL	NULL	Lucca Comix & Games	2018-10-28 00:00:00	2018-10-31 00:00:00	Abb4ggRtd	7.50
Lucca Comix & Games	Lucca	NULL	NULL	Lucca Comix & Games	2018-10-28 00:00:00	2018-10-31 00:00:00	BaseRtd	7.50
Lucca Comix & Games	Lucca	NULL	NULL	Lucca Comix & Games	2018-10-28 00:00:00	2018-10-31 00:00:00	Default	10.00
Lucca Comix & Games	Lucca	NULL	NULL	Gara di cosplay	2018-10-31 00:00:00	2018-10-31 00:00:00	Default	10.00
Carnevale	Ivrea	NULL	NULL	Carnevale	2018-02-08 00:00:00	2018-02-13 00:00:00	Default	10.00
Carnevale	Venezia	NULL	NULL	Carnevale	2018-02-08 00:00:00	2018-02-13 00:00:00	Default	10.00
Fiera campionaria	Padova	Fiera	NULL	Fiera campionaria	2018-05-10 00:00:00	2018-05-18 00:00:00	Default	10.00
Fiera campionaria	Padova	Fiera	P01	Degustazione di cioccolato	2018-05-15 00:00:00	2018-05-16 00:00:00	BaseRtd	7.00
Fiera campionaria	Padova	Fiera	P02	Spritz	2018-05-15 00:00:00	2018-05-16 00:00:00	Default	10.00
Fiera campionaria	Padova	Fiera	NULL	Fiera campionaria	2018-09-20 00:00:00	2018-09-30 00:00:00	Abb3gg	18.00
Fiera campionaria	Padova	Fiera	NULL	Fiera campionaria	2018-09-20 00:00:00	2018-09-30 00:00:00	Default	10.00
Fiera campionaria	Padova	Fiera	P02	Spritz	2018-09-25 00:00:00	2018-09-26 00:00:00	Default	10.00
MantovaComix	Mantova	Fiera	NULL	MantovaComix	2018-08-28 00:00:00	2018-09-03 00:00:00	Default	10.00
MantovaComix	Mantova	Fiera	P03	Gara di cosplay	2018-08-31 00:00:00	2018-08-31 00:00:00	Default	10.00
Fiera del Radioamatore ed Elettronica	Pordenone	Fiera	NULL	Fiera del Radioamatore ed Elettronica	2018-09-20 00:00:00	2018-09-30 00:00:00	Default	10.00
Fiera del Radioamatore ed Elettronica	Pordenone	Fiera	P01	Esposizione radio d'epoca	2018-09-20 00:00:00	2018-09-30 00:00:00	Base	5.00
Fiera del Radioamatore ed Elettronica	Pordenone	Fiera	P01	Esposizione radio d'epoca	2018-09-20 00:00:00	2018-09-30 00:00:00	Default	10.00
Fiera del Radioamatore ed Elettronica	Pordenone	Fiera	P02	Comunicazione Internazionale	2018-09-20 00:00:00	2018-09-30 00:00:00	Base	5.00
Fiera del Radioamatore ed Elettronica	Pordenone	Fiera	P02	Comunicazione Internazionale	2018-09-20 00:00:00	2018-09-30 00:00:00	Default	10.00

Mostra tutte le fiere con i relativi attributi, gli eventi e tariffe associate agli eventi

##Fiere con i relativi eventi, date, padiglioni con Capienza

CREATE VIEW PadiglioniEventi **AS SELECT** Fiera.id **as** IdFiera, Evento.Id **as** IdEvento, Padiglione.Id **as** IdPadiglione, Fiera.Città, Fiera.Luogo, Padiglione.Capienza, Evento.DataInizio, Evento.DataFine, Evento.Nome **as** 'Evento', Fiera.Nome **as** 'Fiera' **FROM** Evento **LEFT JOIN** Fiera **ON** Evento.IdFiera=Fiera.Id **LEFT JOIN** Padiglione **ON** Evento.Padiglione=Padiglione.Id **AND** Fiera.Città=Padiglione.Città **AND** Fiera.Luogo=Padiglione.Luogo **WHERE** !IsNull(Evento.Padiglione);

IdFiera	IdEvento	IdPadiglione	Città	Luogo	Capienza	DataInizio	DataFine	Evento	Fiera
79a4201800	001	P01	Padova	Fiera	500	2018-05-15 00:00:00	2018-05-16 00:00:00	Degustazione di cioccolato	Fiera campionaria
79a4201801	001	P02	Padova	Fiera	300	2018-09-25 00:00:00	2018-09-26 00:00:00	Spritz	Fiera campionaria
cf3e201800	001	P03	Mantova	Fiera	253	2018-08-31 00:00:00	2018-08-31 00:00:00	Gara di cosplay	MantovaComix
e695201800	001	P01	Pordenone	Fiera	500	2018-09-20 00:00:00	2018-09-30 00:00:00	Esposizione radio d'epoca	Fiera del Radioamatore ed Elettronica
79a4201800	002	P02	Padova	Fiera	300	2018-05-15 00:00:00	2018-05-16 00:00:00	Spritz	Fiera campionaria
e695201800	002	P02	Pordenone	Fiera	236	2018-09-20 00:00:00	2018-09-30 00:00:00	Comunicazione Internazionale	Fiera del Radioamatore ed Elettronica

Mostra tutte le fiere con i relativi attributi e gli eventi associati indicandone il luogo le date e la capienza massima dell'evento

PROCEDURE

DELIMITER \$\$

DROP PROCEDURE IF EXISTS PadDisp;

CREATE PROCEDURE PadDisp (**IN** FDataInizio datetime, **IN** FDataFine datetime, **IN** FCittà char(30))

SELECT Id **FROM** Padiglione **WHERE** (Id, Città, Luogo) **NOT IN** (**SELECT** IdPadiglione, Città, Luogo **FROM** PadiglioniEventi

WHERE !(FDataFine < PadiglioniEventi.DataInizio **OR** FDataInizio > PadiglioniEventi.DataFine)) **WHERE** Città=FCittà);

\$\$

DELIMITER ;

```
MariaDB [EzTicket]> call PadDisp('2018-05-15 00:00:00', '2018-05-16 00:00:00', 'Padova');
```

Id	Città	Luogo	Capienza
P01	Padova	Geox	100
P02	Padova	Geox	30
P03	Padova	Fiera	50
P04	Padova	Fiera	500

Mostra i padiglioni disponibili con gli attributi in una determinata città in un lasso di tempo

DELIMITER \$\$

DROP PROCEDURE IF EXISTS MaxCapFiera;

CREATE PROCEDURE MaxCapFiera (**IN** FId char(10))

SELECT Fiera.Nome, Padiglione.Città, Padiglione.Luogo, sum(Capienza) **as** Capienza **FROM** Padiglione, Fiera **WHERE** (Padiglione.Id, Padiglione.Città, Padiglione.Luogo, Fiera.Id) **IN** (**SELECT** Padiglione **as** Id, Città, Luogo, Fiera.Id **as** IdFiera **FROM** Fiera **LEFT JOIN** Evento **ON** Fiera.Id=Evento.IdFiera **WHERE** Fiera.Id=FId **AND** !IsNull(Padiglione));

\$\$

DELIMITER ;

```
MariaDB [EzTicket]> call MaxCapFiera ('79a4201800');
```

Nome	Città	Luogo	Capienza
Fiera campionaria	Padova	Fiera	800

Mostra la capienza massima della fiera di interesse

```

DELIMITER $$
DROP PROCEDURE IF EXISTS VenditeFiera;
CREATE PROCEDURE VenditeFiera (IN Fld char(10))
CREATE VIEW BigliettiVenduti AS Select Fiera.Nome, count(*) AS NumeroBiglietti FROM
Biglietto, Fiera WHERE (IdFiera, Fiera.Nome) in (SELECT Id, Nome FROM Fiera WHERE
NOME=Fld);
$$
DELIMITER ;

```

```

MariaDB [EzTicket]> call VenditeFiera ('79a4201800');
+-----+-----+
| Nome          | NumeroBiglietti |
+-----+-----+
| Fiera campionaria |          22 |
+-----+-----+

```

Mostra tutti i biglietti venduti in una determinata fiera

```

DELIMITER $$
DROP PROCEDURE IF EXISTS PersoneInFiera;
CREATE PROCEDURE PersoneInFiera (IN Fld char(10))
SELECT Fiera.Nome, count(*) AS 'Persone in fiera' FROM Biglietto, Fiera WHERE (IdFiera,
Fiera.Nome) in (SELECT Id, Nome FROM Fiera WHERE NOME=Fld)
AND !IsNull(DataConvalida);
$$
DELIMITER ;

```

```

MariaDB [EzTicket]> call PersoneInFiera('17ed201800');
+-----+-----+
| Nome          | Persone in fiera |
+-----+-----+
| Lucca Comix & Games |          1 |
+-----+-----+

```

Mostra tutti gli ingressi in una determinata fiera

```

DELIMITER $$
DROP PROCEDURE IF EXISTS BigliettiCompratiDaCliente;
CREATE PROCEDURE BigliettiCompratiDaCliente (IN FCF char(16), IN FY int)
SELECT Cliente.Nome, Cliente.Cognome, count(*) as Biglietti_Acquistati FROM Cliente
LEFT JOIN Biglietto ON Cliente.CF=Biglietto.Cliente WHERE CF=FCF AND
YEAR(DEmissione)=FY;
$$
DELIMITER ;

```

```

MariaDB [EzTicket]> call BigliettiCompratiDaCliente('PTTGMR93A28E379G','2018');
+-----+-----+-----+
| Nome      | Cognome   | Biglietti_Acquistati |
+-----+-----+-----+
| Gianmarco | Pettinato |          1 |
+-----+-----+-----+

```

Mostra I biglietti comprati da un determinato cliente in un dato anno

DELIMITER \$\$

DROP PROCEDURE IF EXISTS EventoTop;

CREATE PROCEDURE EventoTop (**IN** FId char(10))

SELECT Evento.Nome, count(*) **as** ingressi **FROM** Biglietto, Evento **WHERE**
Biglietto.IdFiera=FId **AND** Evento.IdFiera=FId **AND** Biglietto.IdEvento=Evento.Id **group by**
(Biglietto.IdEvento) **order by** ingressi **DESC LIMIT 1**;

\$\$

DELIMITER ;

```
MariaDB [EzTicket]> call EventoTop('79a4201800');
+-----+-----+
| Nome   | ingressi |
+-----+-----+
| Spritz |         8 |
+-----+-----+
```

Mostra l'evento di maggior successo relativo alla fiera inserita

DELIMITER \$\$

DROP PROCEDURE IF EXISTS EventoTariffa;

CREATE PROCEDURE EventoTariffa (**IN** FNomeEvento varchar(10))

SELECT * FROM InfoFiere **WHERE** NomeEvento=FNomeEvento;

\$\$

DELIMITER ;

```
MariaDB [EzTicket]> call EventoTariffa('Spritz');
+-----+-----+-----+-----+-----+-----+-----+-----+
| Nome      | Città | Luogo | Padiglione | NomeEvento | DataInizio      | DataFine      | Tipo      | Prezzo |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Fiera campionaria | Padova | Fiera | P02      | Spritz     | 2018-05-15 00:00:00 | 2018-05-16 00:00:00 | Default   | 10.00 |
| Fiera campionaria | Padova | Fiera | P02      | Spritz     | 2018-09-25 00:00:00 | 2018-09-26 00:00:00 | Default   | 10.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Mostra tutti gli eventi con le relative tariffe aventi lo stesso nome di quello inserito

DELIMITER \$\$

DROP PROCEDURE IF EXISTS EventoTempo;

CREATE PROCEDURE EventoTempo (**IN** FDataInizio datetime, **IN** FDataFine datetime)

SELECT Fiera.Nome, Città, Luogo, Padiglione, Evento.Nome,
Evento.DataInizio, Evento.DataFine **FROM** Fiera **LEFT JOIN** Evento **ON**
Fiera.id=Evento.IdFiera **WHERE** FDataInizio>=Evento.DataInizio **AND**
FDataFine<=Evento.DataFine;

\$\$

DELIMITER ;

```
MariaDB [EzTicket]> call EventoTempo('2018-05-15 00:00:00', '2018-05-16 00:00:00');
+-----+-----+-----+-----+-----+-----+-----+
| Nome      | Città | Luogo | Padiglione | Nome      | DataInizio      | DataFine      |
+-----+-----+-----+-----+-----+-----+-----+
| Fiera campionaria | Padova | Fiera | NULL      | Fiera campionaria | 2018-05-10 00:00:00 | 2018-05-18 00:00:00 |
| Fiera campionaria | Padova | Fiera | P01      | Degustazione di cioccolato | 2018-05-15 00:00:00 | 2018-05-16 00:00:00 |
| Fiera campionaria | Padova | Fiera | P02      | Spritz     | 2018-05-15 00:00:00 | 2018-05-16 00:00:00 |
+-----+-----+-----+-----+-----+-----+-----+
```

Mostra tutti gli eventi con le relative informazioni in un lasso di tempo scelto dall'utente

FUNZIONI

```
DELIMITER $$
DROP FUNCTION IF EXISTS CreateFieraId;
CREATE FUNCTION CreateFieraId (FNome varchar(255),FCittà varchar(255),FData DATE)
returns char(10)
BEGIN
    DECLARE BigId char(10);
    DECLARE TempId VARCHAR(10);
    DECLARE Numb int(2);
    SET TempId = CONCAT(CONVERT (MD5(CONCAT(FNome,' ',FCittà)),char(4)),
YEAR(FData));
    SELECT Id FROM Fiera where Id LIKE (CONCAT(TempId,'%')) order by Id DESC LIMIT 1
    INTO BigId;
    IF IsNull(BigId)=1 THEN
        SET TempId = CONCAT(TempId,'00');
    ELSE
        SET Numb = CONVERT ((SELECT RIGHT (BigId,2)), int);
        IF Numb>=99 THEN
            SIGNAL SQLSTATE '45000'
            SET MYSQL_ERRNO=30001,
            MESSAGE_TEXT='Impossibile creare la Fiera con questo nome';
        END IF;
        SET Numb = Numb + 1;
        IF Numb < 10 THEN
            SET TempId = CONCAT(TempId,'0',Numb);
        ELSE
            SET TempId = CONCAT(TempId,Numb);
        END IF;
    END IF;
    RETURN TempId;
END$$
DELIMITER ;
```

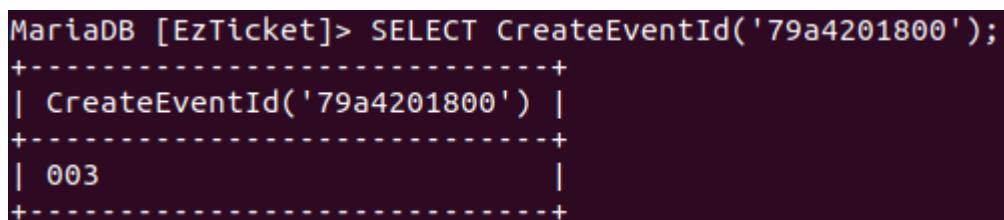
```
MariaDB [EzTicket]> SELECT CreateFieraId('Fiera campionaria','Padova','2018-10-20');
+-----+
| CreateFieraId('Fiera campionaria','Padova','2018-10-20') |
+-----+
| 79a4201802 |
+-----+
```

Partendo dal nome, la città e data la funzione crea un id di 10 caratteri univoco in tutta la base di dati

```

DELIMITER $$
DROP FUNCTION IF EXISTS CreateEventId;
CREATE FUNCTION CreateEventId (FldFiera varchar(10)) RETURNS char(3)
BEGIN
DECLARE LastId char(3);
SET LastId = CONVERT((SELECT Id FROM Evento WHERE Evento.IdFiera=FldFiera
order by Id DESC LIMIT 1), int) + 1;
IF IsNull(LastId) THEN
SET LastId = '000';
ELSEIF LastId < 10 THEN
SET LastId = CONCAT('00',LastId);
ELSEIF LastId < 100 THEN
SET LastId = CONCAT('0',LastId);
END IF;
RETURN LastId;
END$$
DELIMITER ;

```



```

MariaDB [EzTicket]> SELECT CreateEventId('79a4201800');
+-----+
| CreateEventId('79a4201800') |
+-----+
| 003                          |
+-----+

```

Crea un id di 3 caratteri, per un evento non ancora inserito, unico rispettivamente al codice della fiera relativa all'evento stesso

TRIGGER

Inserisce l'id della fiera creato dalla funzione CreateFieraId()

```

DELIMITER ||
DROP TRIGGER IF EXISTS Createld;
CREATE TRIGGER Createld
BEFORE insert on Fiera FOR EACH ROW
BEGIN
SET NEW.Id = CreateFieraId(NEW.Nome,NEW.Città,NEW.DataInizio);
END ||
DELIMITER ;

```

Crea un evento fittizio che rappresenta tutta la fiera

```

DELIMITER ||
DROP TRIGGER IF EXISTS CreatEvent;
CREATE TRIGGER CreatEvent
AFTER insert on Fiera FOR EACH ROW
BEGIN
INSERT INTO `Evento` (`Id`, `IdFiera`, `Nome`, `Padiglione`, `DataInizio`, `DataFine`)
VALUES ('000', NEW.Id, NEW.Nome, NULL, NEW.DataInizio, NEW.DataFine);
END ||
DELIMITER ;

```

Controlla che le date del biglietto siano coerenti con l'evento e che non si superi la capienza massima del padiglione che ospita l'evento stesso

DELIMITER //

DROP TRIGGER IF EXISTS ValidTicket;

CREATE TRIGGER ValidTicket

BEFORE **insert on** Biglietto **FOR EACH ROW**

BEGIN

DECLARE Cgiorni int;

DECLARE NTicket int;

DECLARE NId int(8);

SELECT Id **FROM** Biglietto **WHERE** IdEvento=NEW.IdEvento **AND** IdFiera=NEW.IdFiera
ORDER BY Id **DESC LIMIT 1 INTO** NId;

IF IsNull(NId) **!= 1** **THEN**

SET NEW.Id = NId+1;

ELSE

SET NEW.Id=0;

END IF;

IF ((NEW.DataInizioValidita < (**SELECT** DataInizio **FROM** Evento **WHERE**
Id=NEW.IdEvento **AND** IdFiera=NEW.IdFiera **LIMIT 1**)) **OR**

(NEW.DataInizioValidita > (**SELECT** DataFine **FROM** Evento **WHERE** Id=NEW.IdEvento
AND IdFiera=NEW.IdFiera **LIMIT 1**))) **OR**

IsNull(NEW.DataInizioValidita) **THEN**

SET NEW.DataInizioValidita = (**SELECT** DataInizio **FROM** Evento **WHERE**
Id=NEW.IdEvento **AND** IdFiera=NEW.IdFiera **LIMIT 1**);

END IF;

SELECT count(*) **FROM** Biglietto **WHERE** IdEvento=NEW.IdEvento **AND**
IdFiera=NEW.IdFiera **INTO** NTicket;

SELECT Giorni **FROM** Tariffa **WHERE** IdEvento=NEW.IdEvento **AND**
IdFiera=NEW.IdFiera **AND** Tipo=NEW.Tipo **AND** Prezzo=NEW.Prezzo **LIMIT 1 INTO**
Cgiorni;

SET NEW.DataFineValidita = DATE_ADD(NEW.DataInizioValidita, INTERVAL Cgiorni
DAY);

IF (NEW.DataFineValidita > (**SELECT** DataFine **FROM** Evento **WHERE** Id=NEW.IdEvento
AND IdFiera=NEW.IdFiera **LIMIT 1**)) **THEN**

SET NEW.DataFineValidita = (**SELECT** DataFine **FROM** Evento **WHERE**
Id=NEW.IdEvento **AND** IdFiera=NEW.IdFiera **LIMIT 1**);

END IF;

IF (NTicket>=(**SELECT** Capienza **FROM** PadiglioniEventi **WHERE**
IdEvento=NEW.IdEvento **AND** IdFiera=NEW.IdFiera) **THEN**

SIGNAL SQLSTATE '45000'

SET MYSQL_ERRNO=30002,

MESSAGE_TEXT='Capienza massima superata impossibile creare il biglietto';

END IF;

END //

DELIMITER ;

Controlla che le date dell'evento siano coerenti con quelle della fiera e l'eventuale padiglione scelto sia situati nello stesso luogo della fiera al quale appartiene l'evento stesso(questo controllo viene fatto anche in modifica)

DELIMITER //

DROP TRIGGER IF EXISTS ControlEvent;

CREATE TRIGGER ControlEvent

BEFORE insert on Evento **FOR EACH ROW**

BEGIN

DECLARE DIni datetime;

DECLARE DFin datetime;

DECLARE FLuogo char(30);

DECLARE FCittà char(30);

SELECT Fiera.DataInizio **FROM** Fiera **WHERE** Fiera.Id=NEW.IdFiera **LIMIT 1 INTO** DIni;

SELECT Fiera.DataFine **FROM** Fiera **WHERE** Fiera.Id=NEW.IdFiera **LIMIT 1 INTO** DFin;

SET NEW.Id=CreateEventId(NEW.IdFiera);

IF IsNull(NEW.DataInizio) **THEN**

SET NEW.DataInizio=DIni;

END IF;

IF IsNull(NEW.DataInizio) **THEN**

SET NEW.DataFine=DFin;

END IF;

IF DIni>NEW.DataInizio **OR** DFin<NEW.DataFine **OR** NEW.DataInizio>NEW.DataFine

THEN

SIGNAL SQLSTATE '45000'

SET MYSQL_ERRNO=30003,

MESSAGE_TEXT='Evento non conforme alle date della Fiera';

END IF;

SELECT Città **FROM** Fiera **WHERE** Id = NEW.IdFiera **INTO** FCittà;

SELECT Luogo **FROM** Fiera **WHERE** Id = NEW.IdFiera **INTO** FLuogo;

IF !IsNull(FLuogo) **THEN**

IF !IsNull(NEW.Padiglione) **AND** (NEW.Padiglione **NOT IN** (**SELECT** Id **FROM** Padiglione

WHERE (Id,Città,Luogo) **NOT IN**(**SELECT** IdPadiglione,Città,Luogo **FROM** PadiglioniEventi

WHERE !(NEW.DataFine<PadiglioniEventi.DataInizio **OR**

NEW.DataInizio>PadiglioniEventi.DataFine)) **AND Città=FCittà **AND** Luogo=FLuogo))**

THEN

SIGNAL SQLSTATE '45000'

SET MYSQL_ERRNO=30004,

MESSAGE_TEXT='Padiglione non disponibile';

END IF;

ELSE

IF !IsNull(NEW.Padiglione) **THEN**

SIGNAL SQLSTATE '45000'

SET MYSQL_ERRNO=30005,

MESSAGE_TEXT='Non puoi selezionare un Padiglione';

END IF;

END IF;

END //

DELIMITER ;

Crea una tariffa di Default(modificabile in seguito) per ogni evento creato

DELIMITER //

DROP TRIGGER IF EXISTS CreateTariffa;

CREATE TRIGGER CreateTariffa

AFTER **insert on** Evento **FOR EACH ROW**

BEGIN

DECLARE PrezzoBase DECIMAL(4,2);

SET PrezzoBase='10.00';

INSERT INTO `Tariffa` (`IdFiera`, `IdEvento`, `Tipo`, `Prezzo`, `Giorni`) **VALUES**
(NEW.IdFiera,NEW.Id,'Default',PrezzoBase,'1');

END //

DELIMITER ;

Controlla che quando si aggiorna la data di convalida essa sia coerente con la data di fine validità dello stesso

DELIMITER //

DROP TRIGGER IF EXISTS CheckBiglietto;

CREATE TRIGGER CheckBiglietto

BEFORE **UPDATE on** Biglietto **FOR EACH ROW**

BEGIN

IF NEW.DataConvalida > OLD.DataFineValidita THEN

SIGNAL SQLSTATE '45000'

SET MYSQL_ERRNO=30006,

MESSAGE_TEXT='Il biglietto non è valido';

END **IF**;

END //

DELIMITER ;