

# Progetto di basi

## **1 Abstract**

Il progetto consiste nel modellare una base di dati relativa ad un festival musicale diviso in varie date contigue.

Il Festival prevede diverse esibizioni nell'arco di una giornata su palchi diversi, gli artisti inoltre sono divisi tra gruppi e solisti ed ognuno con un genere musicale.

Le varie giornate, oltre alle LineUp, sono differenziate dal prezzo del biglietto giornaliero.

Il DataBase è destinato all'utenza finale che vuole consultare la LineUp giornaliera, avere informazioni sugli artisti e la posizione della loro esibizione, verificare i punti di ristoro presenti all'interno del Festival.

## 2 Descrizione

Si vuole realizzare una Base di Dati per informare il cliente finale dei servizi di un Festival Musicale.

L'utente di questo Database deve avere la possibilità, in qualsiasi momento, di visualizzare le informazioni aggiornate delle esibizioni con le relative informazioni dei gruppi e/o solisti e dei punti ristoro localizzati nelle vicinanze delle esibizioni.

L'entità principale che si vuole realizzare è, quindi, **esibizione** di cui interessa il palco in cui si esibiscono, il giorno e l'ora e l'ospite di essa.

Ogni ospite può essere un **solista** o un **gruppo musicale**.

Ogni **gruppo musicale** è identificato da un nome, dalla nazione, dal genere, dal numero dei componenti e dalle loro caratteristiche ed infine dall'anno di inizio della loro attività.

Invece ogni **solista** sarà sempre identificato dal nome che può essere diverso dal nome di battesimo.

Deve esserci la possibilità di visualizzare le caratteristiche fondamentali degli elementi dei gruppi e dei solisti quali la data di nascita, nome di battesimo, cognome di battesimo, nazionalità, strumenti musicali che suona e se è cantante.

Ogni **palco** ha almeno un **punto ristoro** situato nelle vicinanze.

Ogni **punto ristoro** vende bevande ma non tutti hanno la possibilità di vendere cibo, può inoltre avere posti a sedere ed essere situato all'interno(indoor) o all'esterno.

## 2 Analisi dei Requisiti

### 2.1 Lista delle Classi

Legenda Attributi(Nome attributo-Tipo Attributo)

1. **Ospite:** modella la generalizzazione di Ospite generico.

Attributi:

- NomeOspite-varchar.
- Nazione-varchar.
- Genere-varchar.
- AnnoInizioAttività-int.

Sono definite le seguenti Sottoclassi:

- 1.1 **Solista:** modella un Ospite Solista.

Attributi:

- PersonaFisica-varchar.

- 1.2 **Gruppo Musicale:** modella un Ospite Gruppo Musicale.

Attributi:

- Componenti-varchar.

2. **Palco:** modella un palco del Festival.

Attributi:

- NomePalco-varchar.
- Capienza-int.
- GrandezzaPalco-smallint.
- PotenzaSonoraDecibel-smallint.
- Indoor-boolean.

**3. Giornata:** modella le descrizioni di una giornata del Festival.

Attributi:

- DataOra inizio-Datetime.
- PrezzoBiglietto-numeric.
- DataOraFine-Datetime

**4. Esibizione:** modella un esibizione di Ospite.

Attributi:

- Ospite-varchar.
- Palco-varchar.
- Giornata-DateTime.
- OraInizio-Time.
- OraFine-Time.

**5. GenereMusicale:** modella un genere musicale.

Attributi:

- NomeGenere-varchar.
- Provenienza-varchar.

**6. PersonaFisica:** modella i dati personale di un ospite solista o di un componente del gruppo.

Attributi:

- CodiceFiscale-varchar.
- NomeBattesimo-varchar.
- CognomeBattesimo-varchar.
- DataNascita-Date.
- Nazionalità-varchar.
- StrumentiSuonati-varchar.
- Cantate-boolean.

**7. StrumentoMusicale:** modella un strumento musicale.

Attributi:

- NomeStrumento-varchar.
- Tipo-varchar.

**8. Punto Ristoro:** modella un punto ristoro.

Attributi:

- Nome-varchar.
- VicinanzaPalco-varchar.
- PostiSedere-smallint.
- Cibo-boolean.
- Indoor-boolean.

## **2.2 Glossario dei termini**

-Persona fisiche: una persona fisica è rappresentata dai suoi caratteri principali quali nome di battesimo, cognome. Essa non rappresenta una persona qualunque ma rappresenta una persona fisica che ha attitudini musicali, cioè che suona almeno uno strumento e/o canta.

-Ospite: un ospite può essere un solista o un gruppo musicale, può avere da 1 a n esibizioni, esso è rappresentato dal nome, il genere musicale (uno ed uno solo), dalla nazione (può essere diversa dalla nazione della/e persone fisiche che lo rappresentano) e dall'anno di inizio attività.

-Genere musicale: è il tipo di musica che suona l'ospite (ad esempio rock, pop, elettronica etc), il genere musicale ha una ed una sola provenienza di origine (che può essere una nazione ma anche un'area geografica più o meno estesa).

-Solista: è un ospite composto da una ed una sola persona fisica.

-Gruppo musicale: è un ospite composto da 2 a n persone fisiche.

-Giornata: Ogni giornata del festival ha un prezzo prefissato, una data ed un ora di apertura ed una data ed un ora di chiusura, le date possono essere lontane tra loro al massimo di 24h per mantenere la correttezza della giornata.

-Esibizione: Qualsiasi evento musicale con durata finita, con uno ed uno solo ospite, un solo palco, una ed una sola giornata, un'ora di inizio ed un'ora di fine.

-Strumento Musicale: è un oggetto con lo scopo di creare un suono, esso ha un nome ed un tipo di appartenenza (ad esempio strumento elettronico, a fiato, etc).

-Indoor: se il locale o palco è situato all'interno di un edificio o al coperto in caso di pioggia.

-Palco: qualsiasi locale o struttura adibita a ospitare un'esibizione, esso non ha posti a sedere(per caratteristica del festival), ha una grandezza misurata in metri quadri che indica la grandezza del pavimento dove l'ospite si esibisce, la capienza cioè la quantità di spettatori in piedi che può ospitare il palco, la potenza sonora misurata in watt dell'impianto sonoro del palco.

-Punto di ristoro: è un qualsiasi locale o struttura adibita a servire bevande e, se ne ha la possibilità, servire cibo. Ogni punto di ristoro ha una vicinanza ad uno ed uno solo palco(in caso di ambiguità si riporterà il palco più vicino), i posti a sedere(se è 0 non ha posti a sedere), e la proprietà di essere indoor o no.

## 2.3 Lista Relazioni

In questo sotto capitolo verranno enunciate le principali relazioni in modo non formale.

### Solista-PersonaFisica:**Appartenenza**

- Un solista appartiene ad una persona fisica.

### GruppoMusicale-PersonaFisica:**Appartenenza**

- Tutti i membri del gruppo appartengono a una persona fisica

### Ospite-GenereMusicale:**Genere di appartenenza**

- Ogni Ospite ha un genere musicale.

### PersonaFisica-StrumentoMusicale:**Strumento Suonato**

- Ogni Persona Fisica suona da 0 a n strumenti.

### PuntoRistoro-Palco:**Vicinanza**

- Ogni punto di ristoro è vicino ad uno ed un solo palco, ogni palco puo essere vicino da 0 a n punti di ristoro.

### Esibizione-Ospite:**Partecipazione**

- Ogni esibizione ha un solo ospite e un ospite partecipa a una sola esibizione

### Esibizione-Giornata:**Programmazione**

- Ogni esibizione avviene in una giornata, una giornata puo' avere da 1 a n esibizioni

### Esibizione-Palco:**Posizionamento**



- Ogni esibizione si svolge su un palco, un palco può avere da 1 a n esibizioni.

1.1

## **2.4 Descrizione della gerarchia**

La classe Ospite è una generalizzazione delle classi Solista e Gruppo. Un gruppo è formato da più persone fisiche, ci sarà quindi una relazione 2 a n con PersonaFisica. Un Solista ha invece una relazione 1-1 con PersonaFisica in quanto un Ospite Solista è formato da una sola persona.

## **3 Schema Concettuale**

## **4 Progettazione Logica**

### **4.1 Ristrutturazione schema ER**

## 4.1 Schema relazionale

**Genere**(NomeGenere,Provenienza)

**Ospite**(Nome,AnnoAttivita,Nazionalita)

**PersonaFisica**(Id,Nome,Cognome,DataNascita,  
Nazionalita,Cantante)

**Composizione**(IdPersona,NomeOspite)

**StrumentoMusicale**(Nome,Tipo)

**CapacitaMusicali**(Persona,Strumento)

**Palco**(Nome,Capienza,Grandezza,Indoor,PotenzaSonora)

**Giornata**(Giorno,PrezzoBiglietto)

**Esibizione**(Ospite,Palco,Giornata,OraInizio,OraFine)

**PuntoRistoro**(Nome,Capienza,Indoor,Plva,Cibo, Vicinanza)

## 5 Implementazione dello schema logico

```
DROP TABLE IF EXISTS Genere;  
CREATE TABLE Genere(  
  NomeGenere varchar(20) PRIMARY KEY,  
  Provenienza varchar(30)  
);
```

```
DROP TABLE IF EXISTS Ospite;  
CREATE TABLE Ospite(  
  Nome varchar(20) PRIMARY KEY,  
  Nazione varchar(30) DEFAULT NULL,  
  AnnoAttivita YEAR(4) DEFAULT NULL,  
  Genere varchar(20) REFERENCES Genere(NomeGenere)  
);
```

```
DROP TABLE IF EXISTS PersonaFisica;  
CREATE TABLE PersonaFisica(  
  Id INT NOT NULL AUTO_INCREMENT,
```

```
Nome varchar(20) NOT NULL,  
Cognome varchar(20) NOT NULL,  
DataNascita DATE,  
Cantante BOOLEAN NOT NULL,  
Nazionalita varchar(30),  
PRIMARY KEY(Id)  
);
```

```
DROP TABLE IF EXISTS Composizione;  
CREATE TABLE Composizione(  
IdPersona INT,  
NomeOspite varchar(20),  
PRIMARY KEY(NomeOspite,IdPersona),  
FOREIGN KEY(NomeOspite) REFERENCES Ospite(Nome),  
FOREIGN KEY(IdPersona) REFERENCES PersonaFisica(Id) ON  
DELETE CASCADE  
);
```

```
DROP TABLE IF EXISTS StrumentoMusicale;  
CREATE TABLE StrumentoMusicale(  
Nome varchar(20) PRIMARY KEY,  
Tipo varchar(20) NOT NULL  
);
```

```
DROP TABLE IF EXISTS CapacitaMusicali;  
CREATE TABLE CapacitaMusicali(  
Persona INT,  
Strumento varchar(20),  
PRIMARY KEY(Persona,Strumento),
```

```
FOREIGN KEY(Persona) REFERENCES PersonaFisica(Id),  
FOREIGN KEY(Strumento) REFERENCES  
StrumentoMusicale(Nome)  
);
```

```
DROP TABLE IF EXISTS Palco;  
CREATE TABLE Palco(  
Nome varchar(20) PRIMARY KEY,  
Capienza SMALLINT,  
Grandezza SMALLINT,  
PotenzaSonora SMALLINT,  
Indoor BOOLEAN NOT NULL  
);
```

```
DROP TABLE IF EXISTS Giornata;  
CREATE TABLE Giornata(  
Giorno DATE PRIMARY KEY,  
PrezzoBiglietto SMALLINT NOT NULL  
);
```

```
DROP TABLE IF EXISTS Esibizione;  
CREATE TABLE Esibizione(  
Ospite varchar(20),  
Palco varchar(20),  
Giornata DATE,  
Orainizio TIME NOT NULL,
```

```
OraFine TIME NOT NULL,  
PRIMARY KEY(Ospite,Palco,Giornata),  
FOREIGN KEY(Ospite) REFERENCES Ospite(Nome),  
FOREIGN KEY(Palco) REFERENCES Palco(Nome),  
FOREIGN KEY(Giornata) REFERENCES Giornata(Giorno)  
);
```

```
DROP TABLE IF EXISTS PuntoRistoro;  
CREATE TABLE PuntoRistoro(  
Nome varchar(20) PRIMARY KEY,  
Capienza SMALLINT NOT NULL,  
Cibo BOOLEAN NOT NULL,  
Plva varchar(13) NOT NULL,  
Indoor BOOLEAN NOT NULL,  
Vicinanza varchar(20) REFERENCES Palco(Nome)  
);
```

## 6 Query, procedure trigger e funzioni

### Query 1:

Query che ritorna id nome e cognome degli artisti nati dal 1 gennaio 1990 con indicato il nome e la capienza del palco in cui suonano, la giornata e l'ora di inizio dell'esibizione

```
SELECT pf.Id, pf.Nome, pf.cognome, pa.nome, pa.Capienza,  
e.Giornata, e.Orainizio  
FROM PersonaFisica as pf JOIN Composizione as c on  
pf.Id=c.IdPersona  
JOIN Ospite as o on c.NomeOspite=o.Nome JOIN Esibizione as e  
ON e.Ospite=o.Nome JOIN Palco as pa ON e.Palco=pa.Nome  
WHERE pf.DataNascita>'1989-12-31'  
ORDER BY pf.Id, pa.Grandezza;
```

Risultato:



```
mysql> SELECT pf.Id, pf.Nome,pf.cognome, pa.nome,pa.Capienza, e.Giornata,e.OraInizio
->
-> FROM PersonaFisica as pf JOIN Composizione as c on pf.Id=c.IdPersona
->
-> JOIN Ospite as o on c.NomeOspite=o.Nome JOIN Esibizione as e ON e.Ospite=o.Nome JOIN Palco as pa ON e.Palco=pa.Nome
->
-> WHERE pf.DataNascita>'1989-12-31'
->
-> ORDER BY pf.Id,pa.Grandezza;
+-----+-----+-----+-----+-----+-----+-----+
| Id | Nome          | cognome | nome      | Capienza | Giornata | OraInizio |
+-----+-----+-----+-----+-----+-----+-----+
| 7  | Edward Christopher | Sheeran | Main Stage | 5000    | 2016-08-15 | 23:15:00 |
| 12 | Nicolas       | Jaar    | Cave Stage | 300     | 2016-08-15 | 23:00:00 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## Query 2:

Query che ritorna i nomi dei punti ristoro che vendono cibo e che si trovano vicino a un palco in cui c'è un'esibizione durante l'ora di cena (19:00-21:00) per ogni giornata del festival, vengono anche tornati il nome dell'artista che si esibisce e il nome del palco

```
SELECT e.Giornata,pr.Nome, p.Nome as 'Vicino al palco',e.Ospite
as 'durante esibizione di'
FROM Esibizione as e JOIN Palco as p ON e.Palco=p.Nome JOIN
PuntoRistoro as pr ON pr.Vicinanza=p.Nome
WHERE pr.Cibo=1 && ( (e.OraInizio<='21:00' &&
e.OraInizio>='19:00') || (e.OraFine<='21:00' &&
e.OraFine>='19:00'))
ORDER BY e.Giornata, pr.Nome;
```

Risultato:

```
mysql> SELECT e.Giornata,pr.Nome, p.Nome as 'Vicino al palco',e.Ospite as 'durante esibizione di'
->
-> FROM Esibizione as e JOIN Palco as p ON e.Palco=p.Nome JOIN PuntoRistoro as pr ON pr.Vicinanza=p.Nome
->
-> WHERE pr.Cibo=1 && ( (e.OraInizio<='21:00' && e.OraInizio>='19:00') || (e.OraFine<='21:00' && e.OraFine>='19:00'))
->
-> ORDER BY e.Giornata, pr.Nome;
+-----+-----+-----+-----+
| Giornata | Nome | Vicino al palco | durante esibizione di |
+-----+-----+-----+-----+
| 2016-08-15 | Main Bar | Main Stage | Mumford and sons |
| 2016-08-15 | StreetFood | Second Stage | Blur |
| 2016-08-16 | StreetFood | Second Stage | Aphex Twin |
| 2016-08-17 | Main Bar | Main Stage | Pink Floyd |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

### Query 3:

Query che ritorna nome e cognome degli artisti che si esibiscono con piu ospiti (es. come solista e come parte di un gruppo) con il numero di ospiti di cui fanno parte

```
SELECT a.Nome,a.Cognome,a.NOspiti
FROM(SELECT a.Nome,a.Cognome, COUNT(*) NOspiti
FROM Composizione AS c JOIN Ospite AS b ON
c.NomeOspite=b.Nome JOIN PersonaFisica AS a ON
c.IdPersona=a.Id
GROUP BY a.Nome, a.Cognome) AS a
WHERE NOspiti>1;
```

Risultato:

```
mysql> SELECT a.Nome,a.Cognome,a.NOspiti FROM(SELECT a.Nome,a.Cognome, COUNT(*) NOspiti
->
-> FROM Composizione AS c JOIN Ospite AS b ON c.NomeOspite=b.Nome JOIN PersonaFisica AS a ON c.IdPersona=a.Id
->
-> GROUP BY a.Nome, a.Cognome) AS a WHERE NOspiti>1;
+-----+-----+-----+
| Nome | Cognome | NOspiti |
+-----+-----+-----+
| Damon | Albarn | 2 |
| Giovanni | Pellino | 2 |
+-----+-----+-----+
2 rows in set (0.01 sec)
```

#### **Query 4: Procedura**

Procedura che dato in input il nome di un genere musicale e un giorno del festival ritorna tutte le esibizioni di quel genere