

# Abstract

Si intende rappresentare una base di dati per la gestione di un festival musicale.

Ogni festival ha un nome, una data di inizio, una data di fine, il numero di giornate ed il proprio tema. Il festival è organizzato in giornate, ciascuna contenente vari concerti e la data effettiva in cui è effettuata. Ogni concerto viene effettuato in una giornata specifica del festival, ha un orario di inizio, un orario di fine, viene eseguito in un palcoscenico e prevede l’esibizione di un artista. Ogni artista può essere di tipo headliner o standard, vanno inoltre memorizzati: nome, numero componenti e genere musicale.

Sono presenti diversi palcoscenici, ciascuno ha un nome, la sua location, la grandezza in metri quadri, il tipo di impianto luci (standard, pro) e la grandezza dell’impianto audio (medio, grande e arena). Sono presenti diversi camerini di tipo Vip o Classic, ogni camerino ha una grandezza espressa in metri quadri, ha la possibilità di avere una serie di servizi (alcuni solo per i camerini Vip), ad esempio il minibar, doccia, un numero variabile di divani, stazione makeup e bagno privato. Un camerino può essere utilizzato da un artista nel corso di una giornata del festival.

È necessario archiviare i seguenti dati per i clienti: nome, cognome, indirizzo di fatturazione ed email. Per accedere ad ogni edizione del festival è necessario acquistare biglietti. I biglietti possono essere di vario tipo (1 giorno, 3 giorni, tutto il festival) a vari prezzi.

Vengono salvati gli acquisti dei biglietti da parte dei clienti, in particolare: la data e l’ora di acquisto, l’importo pagato, il tipo di biglietto e il tipo di consegna del biglietto (fisica, con indirizzo di consegna o digitale). Per ogni edizione del festival sono presenti degli sponsor aventi nome, partita IVA, codice fiscale (se presente) e indirizzo fiscale. Viene inoltre memorizzato lo storico del denaro fornito dallo sponsor per ogni edizione del festival in cui ha partecipato.

# Analisi dei requisiti

## **Descrizione testuale**

Il progetto si concentra sull’organizzazione di **festival**, gestendo quando si tengono gli specifici eventi musicali creati e gestiti ogni qualvolta questi vengano organizzati, in un periodo di riferimento e con un nome univoco.

Per ogni festival si ritiene opportuno memorizzare:

* Il periodo di organizzazione
* Il tema
* Il nome univoco/identificativo

Ciascun festival è organizzato in edizioni, ognuna delle quali ha un proprio tema e in ciascuna di esse si svolgono una serie di concerti in varie città di riferimento. Ad ogni edizione partecipano degli **sponsor**, identificati dai dati fiscali e dallo storico dei pagamenti/indennizzi effettuati

Per partecipare ad un festival occorre acquistare dei **biglietti**, i quali possono essere acquistati sia digitalmente che fisicamente; in questo ultimo caso viene specificato un indirizzo di consegna. In merito ai biglietti si intende memorizzare:

* Il numero di giorni del festival
* Il prezzo
* Il tipo del biglietto

In merito alle prenotazioni, intesi come **acquisti** da parte dell’utente finale, si intende quando sia avvenuta la consegna e presso quale indirizzo di consegna, in riferimento ai **clienti** finali, memorizzati attraverso una serie di dati anagrafici, un indirizzo email di riferimento e un identificatore dato dal portale dell’organizzazione al momento della prima registrazione.

Il festival viene organizzato in una serie di **concerti**, svolti in varie giornate, ciascuno con una data di riferimento. Ogni concerto, con un periodo di performance riportato anche in fase di prenotazione e memorizzato nella base di dati, si svolge su una serie di **palcoscenici**, memorizzando:

* La location di svolgimento
* La grandezza del palcoscenico
* Il tipo di impianto che esso gestisce (audio/luci)

Ad ogni concerto si hanno esibizioni di una serie di **artisti**, i quali possono esibirsi in prima persona oppure essere ospiti d’onore (headliners) in una certa esibizione. Per ogni artista si ha un genere musicale di riferimento, un nome che lo identifica e un numero di componenti.

Ciascun artista ha diritto ad un proprio **camerino**, che viene prenotato dall’entourage in fase di organizzazione. Si intende memorizzare la grandezza ed il tipo di servizio offerto; alcuni di questi possono essere dedicati in maniera esclusiva (VIP).

Ogni camerino è dotato di un certo tipo e numero di **servizi**, capendo in base al tipo di camerino se sono dedicati esclusivamente ai VIP o meno.

## **Glossario dei termini**

| **Termine** | **Descrizione** | **Collegamenti** |
| --- | --- | --- |
| Festival | L’evento tematico tenuto in un certo periodo | Nome, Data\_inizio |
| Giornata | I vari giorni in cui il festival si tiene | Festival |
| Artista | Gruppo o solista che effettua performance in un festival | Concerto |
| Palcoscenico | Il luogo in cui si tiene una certa esibizione | Concerto |
| Concerto | L’oggetto di un’esibizione da parte di un artista | Palcoscenico, Artista, Giornata |
| Camerino | Il luogo prenotato come alloggio da un artista prima di un concerto, con un suo tipo | Artista, Servizio |
| Servizio | Un tipo di attività offerto all’artista per le giornate in cui è in una città per un concerto | Camerino |
| Cliente | Una persona che prenota un biglietto per un’esibizione di un festival | Biglietto per mezzo di Acquisto |
| Biglietto | Biglietto di accesso al Festival | Festival |
| Acquisto | Informazioni relative alla prenotazione da parte di un cliente per un concerto | Entità relazionale tra Cliente e Biglietto |
| Headliner/Standard | Il tipo di artista che si esibisce | Attributo di Artista |
| VIP | Un artista importante a cui è dedicato un certo tipo di servizio e camerino | Attributo di Servizio |
| Fisico/Digitale | Il tipo di consegna del biglietto dopo l’acquisto | Attributi di Acquisto |
| Impianto luci/Impianto audio | Il tipo di impianto installato per un concerto | Attributi di Palcoscenico |
| Sponsor | Chi pubblicizza le edizioni festival, con uno storico di sovvenzioni | Festival |

## **Operazioni tipiche**

Le seguenti operazioni sono definite come statistiche generali da parte degli organizzatori:

| **Tipo di operazione** | **Frequenza** |
| --- | --- |
| Elenco performance per giornata di un'edizione del festival | 500 volte al giorno |
| Numero di biglietti venduti in totale per giornata di un’edizione del festival | Mensilmente |
| I festival con più biglietti venduti per tipo di biglietto | Mensilmente |
| Tipo di servizi offerti per camerino | Settimanalmente |
| Performance per giornata per ogni edizione | Mensilmente |
| Occupazione dei camerini per giornata | 50 volte al giorno |
| Aggiornamento dell’occupazione del camerino | Settimanalmente |

Le seguenti operazioni vengono effettuate con gran frequenza durante la durata del festival e nei periodi antecedenti. Alcune di queste operazioni vengono effettuate dal sito internet per la visualizzazione dei dati e l’acquisto da parte del cliente.

| **Tipo di operazione** | **Frequenza** |
| --- | --- |
| Artisti per ogni giornata ed edizione | 1000 volte al giorno (utilizzato nel sito internet) |
| Creazione di cliente | 1000 volte al giorno (form del sito internet) |
| Creazione di un acquisto da parte del cliente | 1000 volte al giorno (form del sito internet) |
| Inserimento di un nuovo artista | 100 volte mensilmente |
| Inserimento del concerto, dove si svolge, in che giornata e che artista si esibisce | Settimanalmente |

# Progettazione concettuale

**Entità**

* **Festival**

1. Nome - Identificativo di un festival | *VARCHAR(100) UNIQUE*
2. Data inizio - Attributo temporale di inizio svolgimento festival | *DATE*
3. Data fine - Attributo temporale di fine svolgimento festival | *DATE*
4. Tema - L’argomento specifico di un’edizione di festival | *VARCHAR(200)*

Chiave primaria: (Nome, Data inizio)

* **Sponsor**

1. Nome - Identificativo semantico di uno sponsor | *VARCHAR(100) UNIQUE*
2. Partita\_iva- Identificativo dell’impresa pubblicitaria | *VARCHAR(11)*
3. Indirizzo fiscale - Locazione fiscale a cui fa capo l’impresa | *VARCHAR(100)*
4. Codice fiscale - Identificativo per singolo dell’impresa pubblicitaria | *VARCHAR(13)*

Chiave primaria: (Partita IVA)

* **Giornata** (Data\_effettiva, Festival)

1. Data effettiva - Quando effettivamente si svolge il festival | *DATE*
2. Festival - La manifestazione a cui si riferisce la giornata | *VARCHAR(100)*

Chiave primaria: (Data effettiva)

Vincoli:

Giornata.Festival → Festival.Nome

* **Artista**

1. Nome - L’artista a cui si fa riferimento | *VARCHAR(100)*
2. Numero\_componenti- L’eventuale numero di artisti dell’entità | *SMALLINT*
3. Genere\_musicale- Il genere di musica proposta dall’artista | *VARCHAR(100)*

L’entità Artista ha due sottocategorie accorpate con una generalizzazione totale: Headliner e Standard. Ciò viene indicato dall’attributo successivo

1. Headliner - Flag che indica se l’artista sia o meno un headliner | *BOOL*

Chiave primaria: (Nome)

* **Palcoscenico**

1. Nome - Identificativo di un palcoscenico| *VARCHAR(50)*
2. Grandezza - Dimensione in metri quadri del camerino | *FLOAT*
3. Location - La città di riferimento del palco | *VARCHAR(100)*
4. Impianto\_luci- Lo specifico impianto luci del palco; viene descritto da apposita struttura *ENUM ('standard', 'pro')*
5. Impianto\_audio - Lo specifico impianto audio del palco; viene descritto da apposita struttura *ENUM (‘medio’, ‘grande’, ‘arena’)*

Chiave primaria: (Nome)

* **Concerto**

1. Giornata - Quando avviene un determinato concerto | *DATE*
2. Artista - Il singolo o il gruppo che si esibisce in un certo momento | *VARCHAR(100)*
3. Ora\_inizio - Momento di inizio del concerto | *TIME*
4. Ora\_fine- Momento di fine del concerto | *TIME*
5. Palcoscenico - Palco di riferimento del concerto in una città | *VARCHAR(50)*

Chiave primaria: (Giornata, Artista)

* **Camerino**

1. ID - Identificativo di un camerino | *VARCHAR(5)*
2. Grandezza - Dimensione in metri quadri del camerino | *FLOAT*

L’entità Camerino ha due sottocategorie accorpate con una generalizzazione totale: VIP e Classic. Essa viene rappresentata dal successivo campo Tipo

* **Servizio**

1. Nome - Il servizio offerto per camerino | *VARCHAR(50)*
2. Solo\_vip - Flag che indica se il servizio si offre solo ai VIP | *BOOL*

Chiave primaria: (Nome)

* **Cliente**

1. ID - Identificativo di un cliente | *VARCHAR(20)*
2. Nome - Nome proprio di riferimento dell’utente | *VARCHAR(100)*
3. Cognome - Cognome di riferimento dell’utente | *VARCHAR(100)*
4. Indirizzo\_fiscale - Indirizzo fiscale di riferimento dell’utente | *VARCHAR(100)*
5. Email - Mezzo di contatto e di registrazione dell’utente finale | *VARCHAR(100)*

Chiave primaria: (ID)

* **Biglietto**

1. ID - Identificativo univoco di un biglietto | *VARCHAR(20)*
2. Festival - La manifestazione di riferimento per una prenotazione | *VARCHAR(100)*
3. Prezzo - Somma di acquisto per una prenotazione | *FLOAT*
4. Numero giorni - Il numero di giorni prenotati per la visione | *SMALLINT*

L’entità Biglietto ha due sottocategorie accorpate con una generalizzazione totale: Fisico e Digitale. Ciò viene indicato dai due attributi successivi.

1. Fisico - Flag che indica se il biglietto sia fisico | *BOOL*
2. Digitale - Flag che indica se il biglietto sia digitale| *BOOL*

Chiave primaria: (ID)

* **Acquisto**

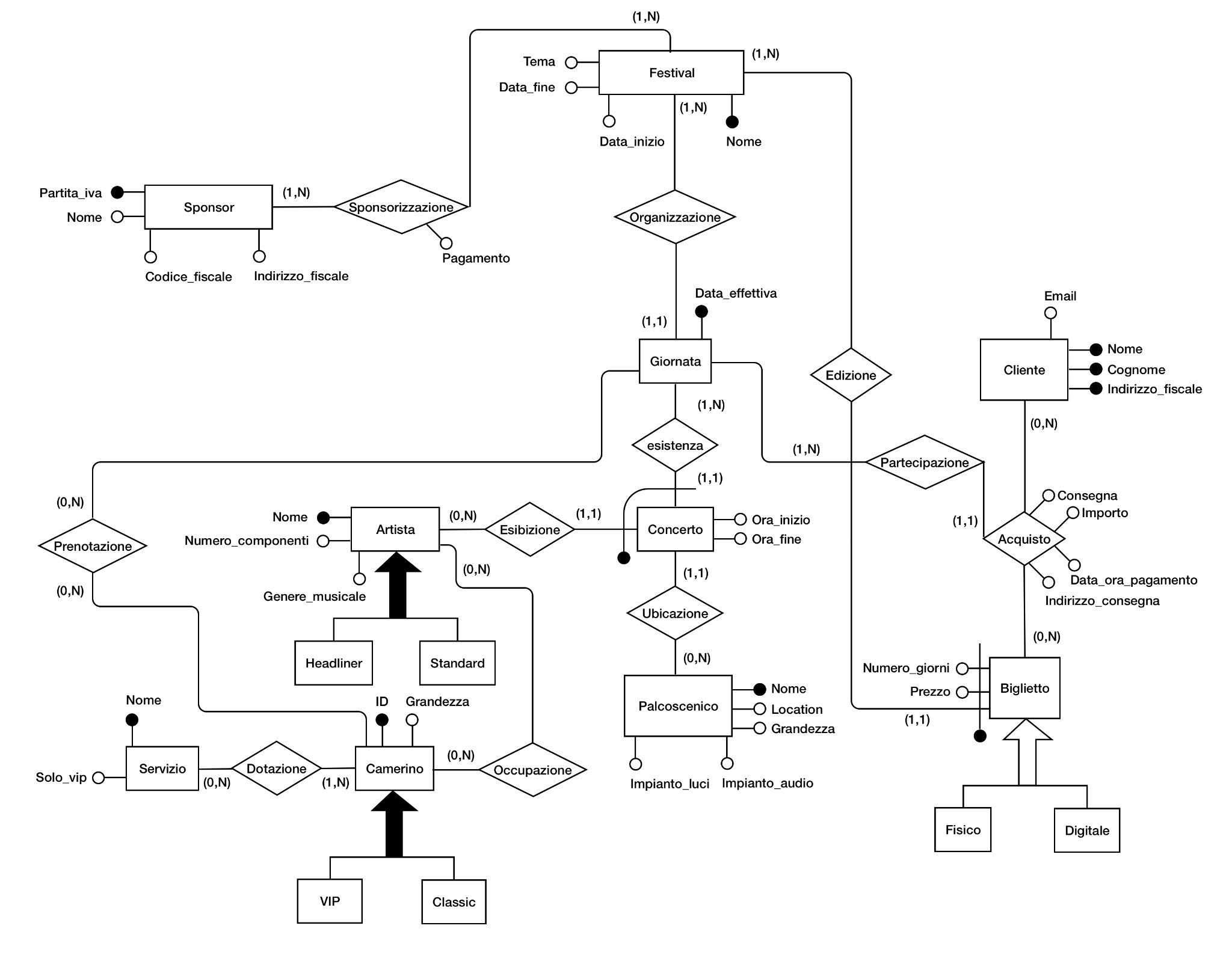
1. ID - Identificativo univoco di acquisto | *VARCHAR(20)*
2. Data\_ora - Il singolo o il gruppo che si esibisce in un certo momento | *TIMESTAMP*
3. Cliente - Momento di inizio del concerto | *TIME*
4. Indirizzo consegna - L’indirizzo indicato a seguito di acquisto di biglietto fisico | *VARCHAR(100)*
5. Importo - Prezzo speso per la compravendita | *FLOAT*
6. Consegna - Il tipo di consegna richiesto in fase di prenotazione; viene descritto da apposita struttura *ENUM (‘digitale’, ‘fisica’, ‘entrambe’)*

Chiave primaria: (ID)

## **Relazioni e cardinalità**

| **Relazione** | **Entità coinvolte** | **Descrizione** |
| --- | --- | --- |
| Sponsorizzazione | Sponsor (1,N)  Festival (1,N) | Uno sponsor pubblicizza più festival. Un’edizione di festival viene sovvenzionata da più sponsor. |
| Organizzazione | Festival (1,N)  Giornata (1,1) | Un festival si svolge in N giornate. Una giornata fa riferimento ad un festival specifico. |
| Edizione | Festival (1,N)  Biglietto (1,1) | Un biglietto fa riferimento ad una specifica edizione del festival. I festival hanno una serie di biglietti. |
| Esistenza | Giornata (1,N)  Concerto (1,1) | Un concerto si svolge in una specifica giornata. In una giornata si svolge una serie di concerti. |
| Acquisto\_biglietto | Acquisto (1,N)  Biglietto (0,N) | Un biglietto può essere acquistato da un cliente. Un acquisto di un cliente si riferisce ad almeno un biglietto. |
| Proprietà | Acquisto (1,1)  Cliente (0,N) | L’acquisto è riferimento ad un cliente univocamente.  Il cliente può effettuare una serie di acquisti. |
| Ubicazione | Concerto (1,1)  Palcoscenico (0,N) | Un concerto si svolge su un certo palco. Su un palcoscenico può svolgersi una serie di concerti. |
| Esibizione | Artista (0,N)  Concerto (1,1) | Un artista può fare una serie di concerti. Un concerto ha come riferimento un artista. |
| Occupazione | Artista (0,N)  Camerino (0,N) | Un artista può occupare una serie di camerini ed un camerino può essere occupato da una serie di artisti. |
| Prenotazione | Camerino (0,N)  Giornata (0,N) | Un camerino può essere prenotato per una serie di giornate. In una giornata può essere prenotata una serie di camerini. |
| Dotazione | Servizio (0,N)  Camerino (1,N) | Un servizio può avere una serie di camerini. Un camerino ha una serie di servizi. |
| Partecipazione | Giornata (1,N)  Acquisto (1,1) | Un acquisto permette di partecipare ad una giornata. Una giornata possiede una serie di acquisti ad essa dedicati |

## Schema concettuale

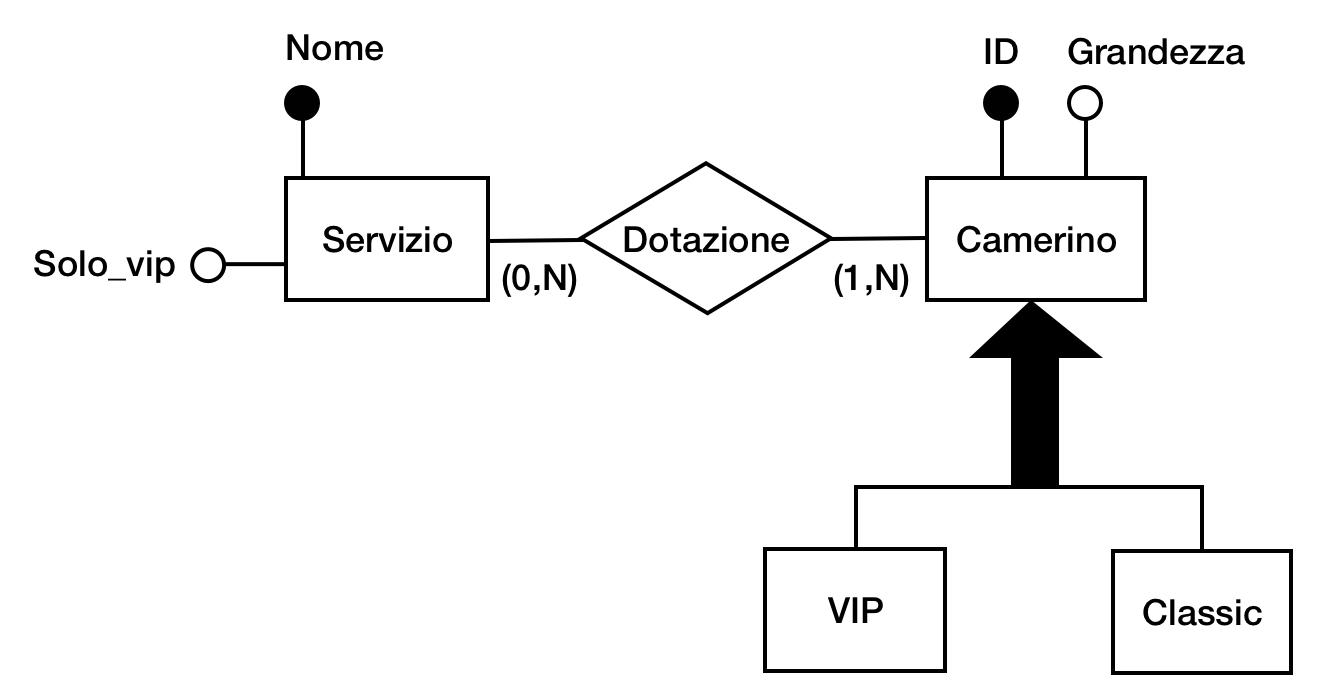


Vincoli non esprimibili in ER

* Ogni cliente può partecipare al massimo al numero di giornate indicate nel tipo di biglietto, e può scegliere a quali giornate partecipare
* Viene fatto un controllo sul biglietto all’ingresso ma non viene archiviato su questo database

# Progettazione logica

## **Analisi delle ridondanze e ristrutturazione**

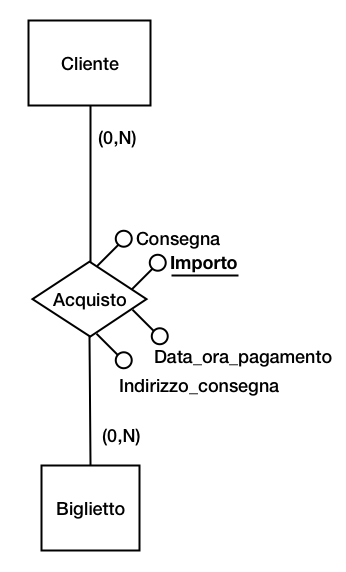
****

Nella **relationship Camerino** è presente una generalizzazione che rende il camerino di tipo Vip o Classic, ad ogni servizio vengono associati diversi servizi, aventi anch’essi la stessa distinzione.

È possibile eliminare direttamente la generalizzazione e presumere che un camerino è di tipo Vip solo se ha associato ad esso almeno un servizio Vip, eliminando quindi ridondanza e possibilità di errori.

Questa informazione viene utilizzata durante l’assegnazione dei camerini ad un artista, operazione che si svolge una volta per artista prima dell’inizio del festival.

Il fatto di trasformare la generalizzazione in un attributo della relazione Camerino non comporta aumento di performance significativo, bensì aumenterebbe la presenza di incongruenze nei dati memorizzati.



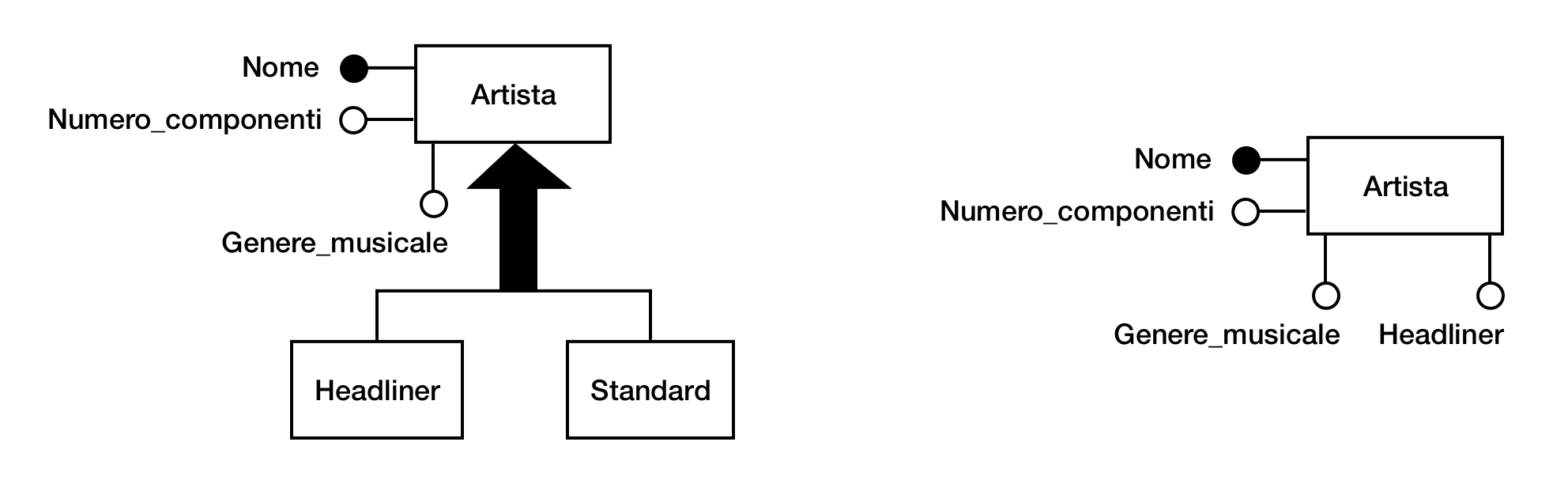
Inoltre nella **relationship Acquisto** tra le entità Biglietto e Cliente si nota la presenza dell’attributo Importo che segna l’importo pagato.

Questo valore può essere ricavato sommando il totale dei biglietti associati all’acquisto.

Visto che il totale dell’importo non è calcolato spesso, la presenza dell’attributo non è giustificata, abbiamo dunque deciso di eliminarlo anche per evitare incongruenze.

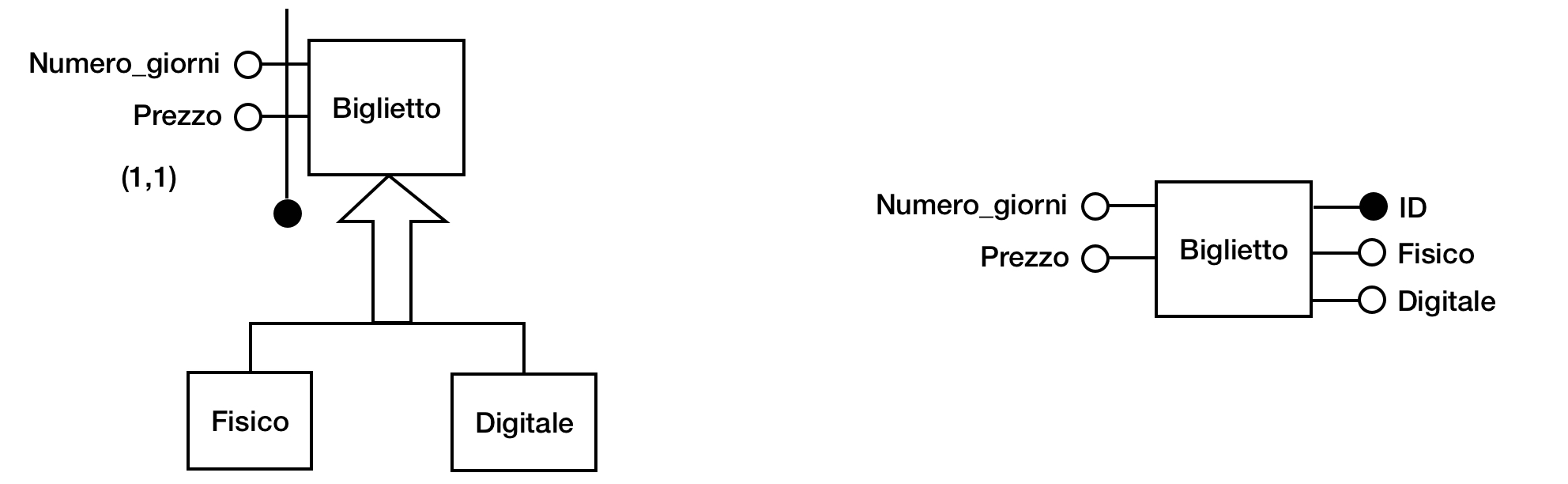
## **Eliminazione delle generalizzazioni**

Di seguito si elencano le generalizzazioni e come sono state rimosse.

Artista 

La relazione Artista presenta una generalizzazione totale in Headliner e Standard, questa può essere trasformata in un campo boolean Headliner che indica se l’artista è Headline o meno.

Biglietto



La relazione Biglietto presenta una generalizzazione parziale in Fisico e Digitale, questa indica se il tipo di biglietto può essere consegnato in maniera fisica (cartacea) e/o digitale (via email).

Essendo una relazione parziale è necessario avere dei valori per biglietti solo fisici, solo digitali ed entrambi. È stato scelto di utilizzare due campi booleani per i tipi di biglietto per chiarezza.

Camerino

La generalizzazione per la relazione Camerino è stata rimossa come discusso in precedenza.

**Scelta degli identificatori primari**

Cliente, Acquisto e Biglietto

Si è scelto di utilizzare un id di 20 caratteri alfanumerici generati casualmente, questo per evitare ID sequenziali e migliorare la sicurezza del database e dell’applicazione che lo utilizza.

Dato le relationships presenti Cliente e Acquisto e tra Acquisto e Biglietto si è manifestata la necessità di creare un ID per le relazioni in modo da rendere la tabella Acquisto più semplice, senza includere tutti gli attributi superchiave di Cliente (Nome, Cognome e Indirizzo) e di Biglietto (Edizione del festival, Numero di Giorni e Prezzo). Questo ragionamente è stato poi esteso alla relazione Acquisto per semplicità e facilità di manutenzione del Database.

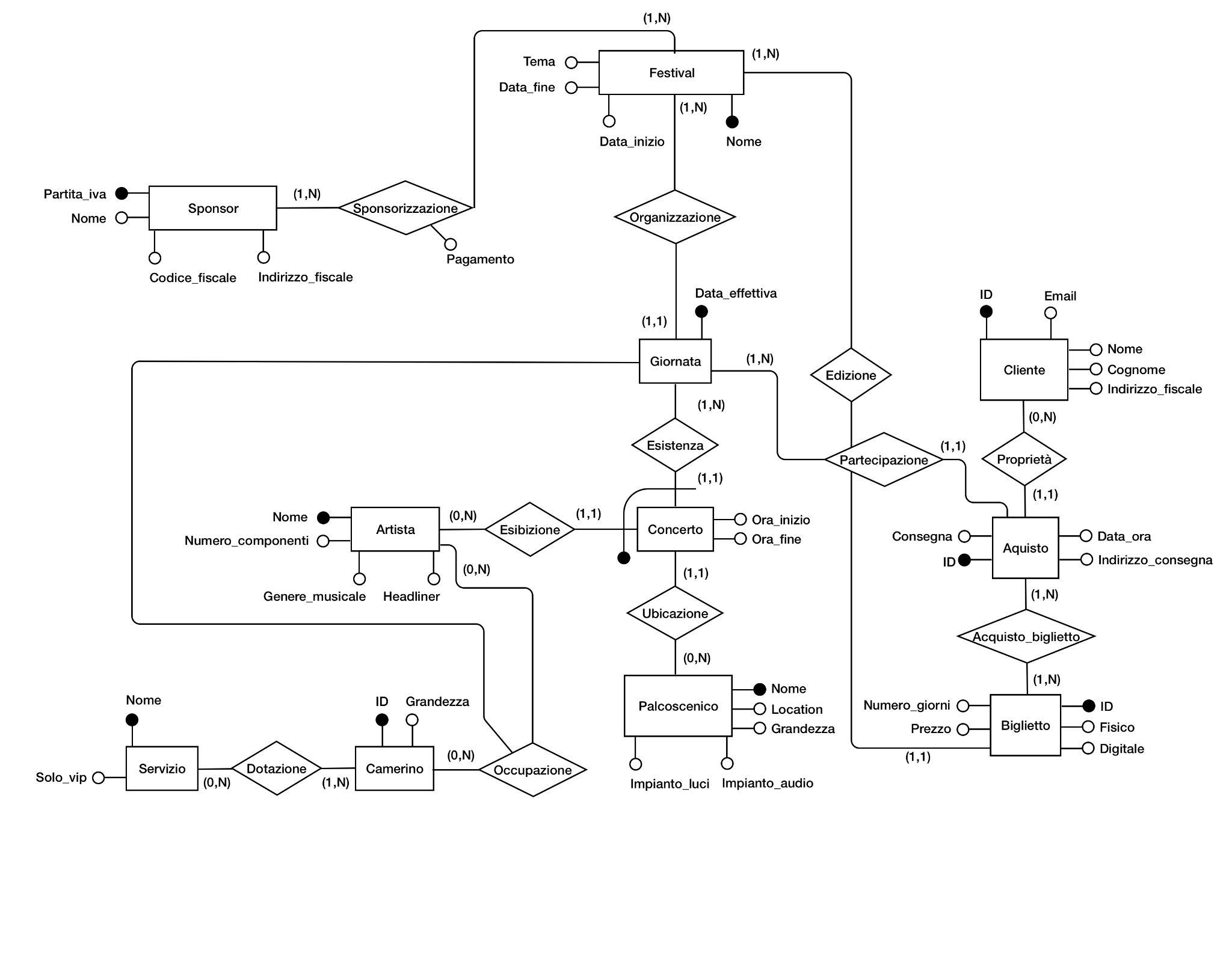
Camerino

Ogni camerino ha un id di 5 caratteri alfanumerici per identificarlo, questo è stato scelto in quanto i camerini non hanno un nome associato e vengono identificati con la lettera C ed un numero non sequenziale (es. C0001, C0020, C1001).

Altre relazioni

Le altre relazioni presenti mantengono come identificatore primario gli attributi che naturalmente li definiscono (es. Il nome o la data in cui avvengono).

## Diagramma E-R ristrutturato



## Schema relazionale e vincoli di integrità referenziale

Si indica con l’asterisco la presenza di valori nulli, con le frecce la presenza di vincoli di integrità referenziale, le sottolineature indicano le chiavi (primarie ed esterne)

* **Festival** (Nome, Data\_inizio, data\_fine, Tema)
* **Sponsor** (Nome, Partita\_iva, Indirizzo\_fiscale, Codice\_fiscale)
* **Sponsorizzazione** (Festival, Pagamento, Sponsor)

Sponsorizzazione.Festival → Festival.Nome

Sponsorizzazione.Sponsor → Sponsor.Partita\_iva

* **Artista** (Nome, Numero\_componenti, Genere\_musicale, Headliner)
* **Giornata** (Data\_effettiva, Festival)

Giornata.Festival → Festival.Nome

* **Palcoscenico** (Nome, Location, Grandezza, Impianto\_luci, Impianto\_audio)
* **Concerto** (Giornata, Artista, Ora\_inizio, Ora\_fine, Palcoscenico)

Concerto.Palcoscenico → Palcoscenico.Nome

Concerto.Artista → Artista.Nome

Concerto.Giornata → Giornata.Data\_effettiva

* **Camerino** (ID, Grandezza)
* **Occupazione** (Artista, Camerino, Giornata)

Occupazione.Artsita → Artista.Nome

Occupazione.Camerino → Camerino.ID

Occupazione.Giornata → Giornata.Data\_effettiva

* **Servizio** (Nome, Solo\_vip)
* **Camerino\_servizio** (Camerino, Servizio)

Camerino\_Servizio.Camerino → Camerino.ID

Camerino.Servizio → Servizio.Nome

* **Cliente** (ID, Nome, Cognome, Indirizzo\_fiscale, Email)
* **Biglietto** (ID, Festival, Prezzo, Numero\_giorni, Fisico, Digitale)

Biglietto.Festival → Festival.Nome

* **Acquisto** (ID, Data\_ora, Cliente, Indirizzo\_consegna, Consegna, Cliente)

Acquisto.Cliente → Cliente.ID

* **Acquisto\_blglietto** (Acquisto, Biglietto)

Acqusto\_biglietto.Acquisto → Acquisto.ID

Acquisto\_biglietto.Biglietto → Biglietto.ID

* **Partecipazione** (Giornata, Acquisto)

Partecipazione.Giornata → Giornata.Data\_effettiva

Partecipazione.Acquisto → Acquisto.ID

## Query SQL - Operazioni tipiche

**SELECT** giornata.data\_effettiva, concerto.artista, concerto.ora\_inizio, concerto.ora\_fine, concerto.palcoscenico

**FROM** giornata,concerto

**WHERE** giornata.festival = 'Fog Festival 2020' **AND** concerto.giornata = giornata.data\_effettiva

**ORDER BY** giornata.data\_effettiva, concerto.ora\_inizio;

**SELECT** giornata.data\_effettiva **AS** giornata, COUNT(\*) **AS** numero\_biglietti

**FROM** partecipazione, acquisto\_biglietto, giornata

**WHERE** giornata.festival = 'Fog Festival 2020'

**AND** giornata.data\_effettiva = partecipazione.giornata

**AND** partecipazione.acquisto = acquisto\_biglietto.acquisto

**GROUP BY** giornata.data\_effettiva

**ORDER BY** giornata.data\_effettiva;

**SELECT** biglietto.festival, **COUNT(\*)** **AS** biglietti

**FROM** acquisto\_biglietto,biglietto

**WHERE** acquisto\_biglietto.biglietto = biglietto.id

**GROUP BY** biglietto.festival

**ORDER BY** **COUNT(\*) DESC**;

**SELECT** camerino.id, camerino\_servizio.servizio

**FROM** camerino, camerino\_servizio

**WHERE** camerino.id = camerino\_servizio.camerino

**ORDER** BY camerino.id

**SELECT** giornata.festival,giornata.data\_effettiva, concerto.artista, concerto.ora\_inizio, concerto.ora\_fine

**FROM** concerto, giornata

**WHERE** concerto.giornata = giornata.data\_effettiva

**ORDER BY** (giornata.festival,giornata.data\_effettiva,concerto.ora\_inizio);

**SELECT** camerino, artista

**FROM** occupazione

**WHERE** giornata = '2020-06-15';

**UPDATE** occupazione **SET** artista='Yellow Floyd'

**WHERE** giornata='2020-06-16' **AND** camerino='CAM03';

Fine tipiche

## Query SQL e Indice TODO: Screen da Postgre e da C++

1 - Mostrare Il festival che ha ricevuto maggiore sponsorizzazione

**CREATE VIEW** sponsorizzazioni\_totali\_festival(festival,totale) AS

(

**SELECT** S1.festival , SUM(S1.storico\_pagamenti) AS fondi\_totali

**FROM** sponsorizzazione S1

**GROUP BY** S1.festival

);

**SELECT** festival, totale

**FROM** sponsorizzazioni\_totali\_festival

**WHERE** totale >= (**SELECT** **MAX**(totale) **FROM** sponsorizzazioni\_totali\_festival)

2 - Mostrare gli artisti headliners del festival con più partecipazione media in base alle giornate

**CREATE VIEW** media\_biglietti\_festival AS

**SELECT** festival, AVG(biglietti) as media\_biglietti\_giornata

**FROM** (

**SELECT** giornata.festival, giornata.data\_effettiva, **COUNT**(\*) as biglietti

**FROM** partecipazione, giornata, acquisto\_biglietto

**WHERE** partecipazione.giornata = giornata.data\_effettiva

**AND**  acquisto\_biglietto.acquisto = partecipazione.acquisto

**GROUP BY** giornata.data\_effettiva

) **AS** biglietti\_giornata

**GROUP BY** festival;

**SELECT** artista.nome

**FROM** artista,giornata,concerto

**WHERE** concerto.giornata = giornata.data\_effettiva

**AND** artista.nome = concerto.artista

**AND** artista.headliner = true

**AND** giornata.festival = (

**SELECT** festival

**FROM** media\_biglietti\_festival

**WHERE** media\_biglietti\_giornata >= (

**SELECT MAX**(media\_biglietti\_giornata) **FROM** media\_biglietti\_festival

)

)

3 - Prezzo medio pagato dai clienti paganti per edizione di un certo festival (in questo esempio Fog Festival 2020)

**SELECT AVG**(prezzo)

**FROM** biglietto, acquisto\_biglietto, acquisto

**WHERE** biglietto.id = acquisto\_biglietto.biglietto

**AND** acquisto\_biglietto.acquisto = acquisto.id

**AND** biglietto.festival = 'Fog Festival 2020'

4 - Artisti non headliner che hanno soggiornato nel camerino VIP più grande e più frequentato

​​

**SELECT** artista.nome

**FROM** occupazione, camerino, artista

**WHERE** occupazione.camerino = camerino.id

**AND** occupazione.artista = artista.nome

**AND** artista.headliner = false

**AND** camerino.grandezza = (

**SELECT** MAX(grandezza)

**FROM** camerino, camerino\_servizio, servizio

**WHERE** camerino.id = camerino\_servizio.camerino

**AND** servizio.nome = camerino\_servizio.servizio

**AND** servizio.solo\_vip = true

**GROUP BY** camerino.id

**LIMIT 1**

)

**GROUP BY** artista.nome, occupazione.camerino;

5 - Numero di concerti effettuati per le combinazioni di tipologia di impianto audio e luci

**SELECT** palcoscenico.impianto\_audio, palcoscenico.impianto\_luci,

**COUNT(\*)** **AS** numero\_concerti

**FROM** concerto, palcoscenico

**WHERE** concerto.palcoscenico = palcoscenico.nome

**AND CONCAT**(palcoscenico.impianto\_audio, palcoscenico.impianto\_luci) IN

(

**SELECT CONCAT**(P1.impianto\_audio,P1.impianto\_luci)

**FROM** palcoscenico P1, palcoscenico P2

**WHERE** P1.impianto\_audio != P2.impianto\_audio

**AND** P1.impianto\_luci != P2.impianto\_luci

)

**GROUP BY CONCAT**(palcoscenico.impianto\_audio, palcoscenico.impianto\_luci), palcoscenico.impianto\_audio, palcoscenico.impianto\_luci

## **Indice**

**CREATE INDEX** Acquisto\_idx **ON** Acquisto(ID);

Essendo un database implementato per un ampio riutilizzo, la parte interessante e che prevede molte operazioni in scrittura, circa 1000 in scrittura e altrettante in lettura, dato che i clienti hanno la possibilità di poter acquistare biglietti prima e durante il festival da sito e da biglietteria.

Si nota quindi che tale operazione deve richiedere una certa velocità per essere eseguita, dunque l’utilizzo opportuno ricade sull’attributo *ID* della tabella *Acquisto*, utile quindi a ricavare contemporaneamente i dati relativi all’edizione del festival (tabella *Festival*) e relativi ai clienti registrati (tabella *Clienti*), i quali hanno la propria partecipazione attraverso un tipo di biglietto (tabella *Biglietti*).

## Codice C++ TODO

TODO:

* Popolamento delle tabelle con API - Da inserire nel file SQL
* Check sulle query struttura
* C++