



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	Attributi di Acquisto

	dopo l'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 - Attributo multivalore poi scomposto nella relazione Indirizzo
 - 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

- 4) 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 - Attributo multivalore poi scomposto nella relazione Indirizzo
- 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.

- 5) Fisico - Flag che indica se il biglietto sia fisico | *BOOL*
- 6) 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 - Attributo multivalore poi scomposto nella relazione Indirizzo
- 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')*

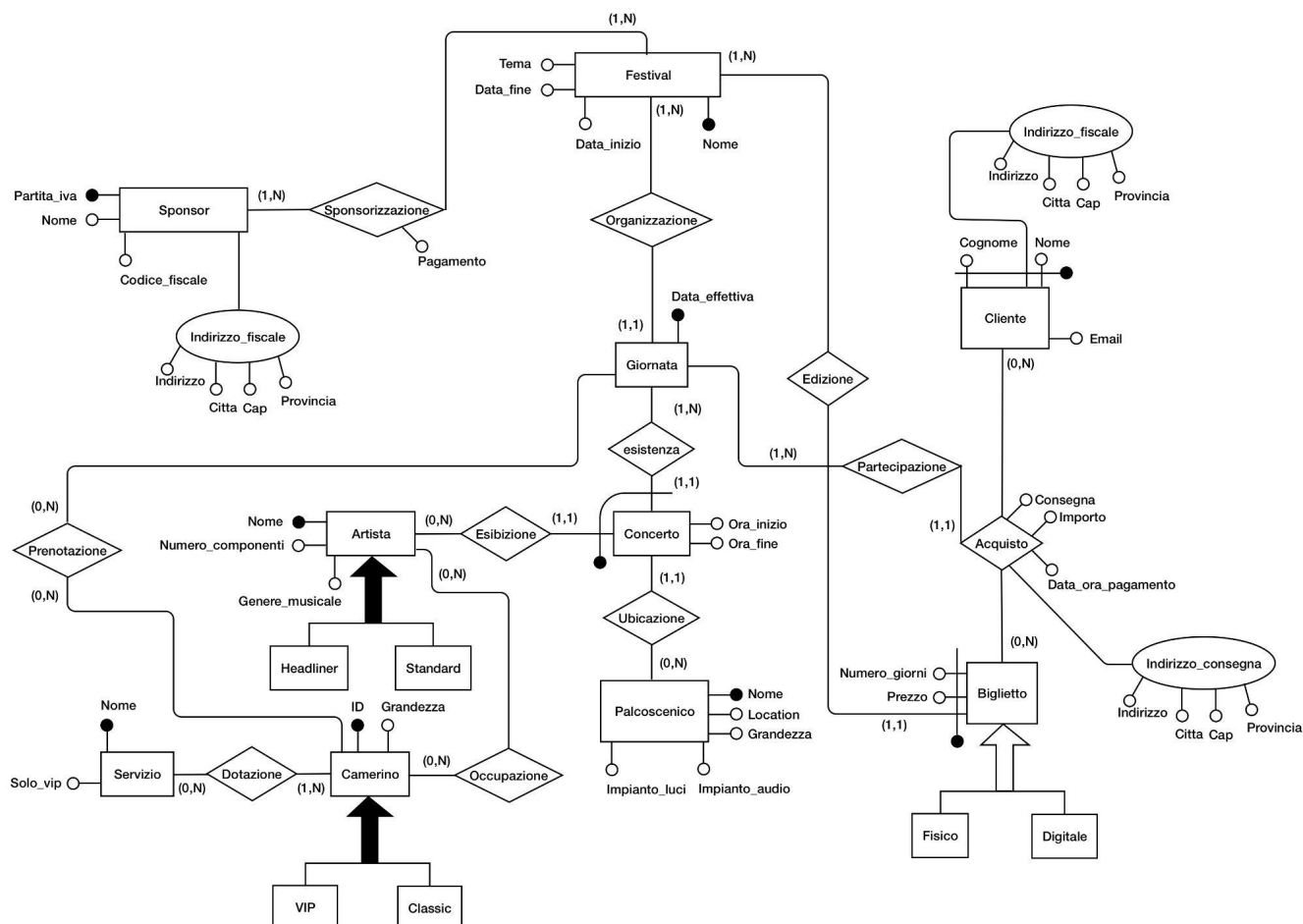
Chiave primaria: (ID)

- **Relazioni e cardinalità**

Relazione	Entità coinvolte	Descrizione
Sponsorizzazione	Sponsor (1,N) Festival (1,N)	Uno sponsor pubblica 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

Creazione della relazione Indirizzo

È stato scelto di creare una relazione chiamata Indirizzo per raccogliere gli indirizzi associati a Sponsor, Clienti e all'Acquisto.

Visto la presenza degli stessi tipi di valore nei tre attributi delle relazioni sopra elencati si è ritenuto opportuno porli in un'unica relazione che poi viene collegata da relationships alle varie entità.

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 ragionamento 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).

Indirizzo

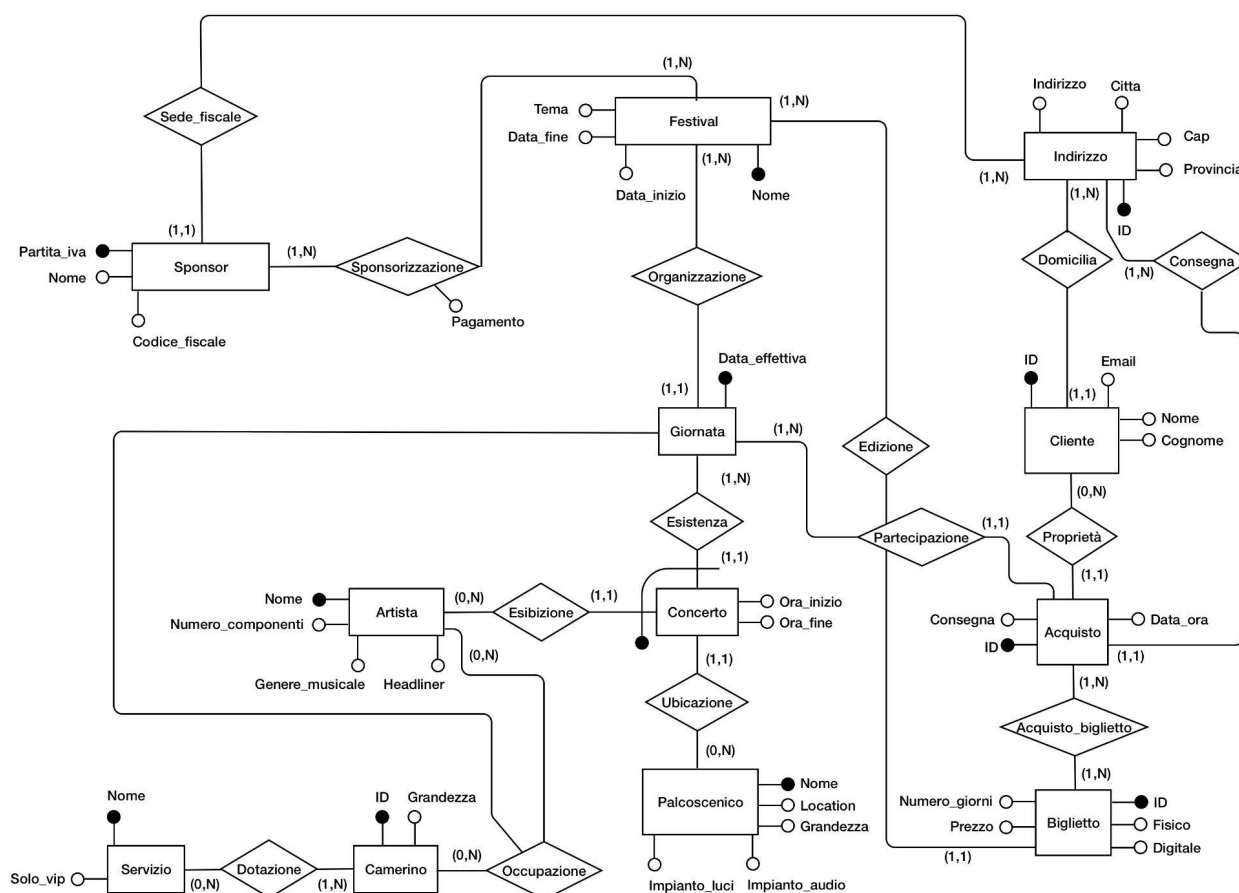
Si è scelto di aggiungere un ID di 20 caratteri alfanumerici per la relazione Indirizzo, in quanto questo presenta relationship con le relazioni Sponsor, Cliente e Acquisto. Inoltre per semplicità e facilità di manutenzione del Database.

L'ID inizia con la lettera S per indirizzi associati a Sponsor, C per indirizzi associati a Clienti e A per indirizzi associati ad acquisti.

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)
- **Indirizzo** (ID, Indirizzo, Citta, Cap, Provincia)
- **Sponsor** (Nome, Partita_iva, Indirizzo_fiscale, Codice_fiscale)
Sponsor.indirizzo_fiscale → Indirizzo(ID)
- **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)
Cliente.indirizzo_fiscale → Indirizzo(ID)
- **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
Sponsor.indirizzo_consegna → Indirizzo(ID)
- **Acquisto_biglietto** (Acquisto, Biglietto)
Acquisto_biglietto.Acquisto → Acquisto.ID
Acquisto_biglietto.Biglietto → Biglietto.ID
- **Partecipazione** (Giornata, Acquisto)
Partecipazione.Giornata → Giornata.Data_effettiva
Partecipazione.Acquisto → Acquisto.ID

Query SQL e Indice

1 - Mostrare il festival che ha ricevuto maggiore sponsorizzazione

```
CREATE VIEW sponsorizzazioni_totali_festival(festival,totale) AS
(
    SELECT S1.festival , SUM(S1.pagamento) 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);

	festival character varying (100)	totale double precision
1	Fog Festival 2020	36300

festival	totale
Fog Festival 2020	36300

2 - Mostrare gli artisti headliner 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 DISTINCT artista.nome AS artista
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
    ));
```

	artista character varying (100)
1	Autotone
2	Elvis Juice
3	Enraged Miss
4	Equatorial Maximum
5	Equatorial Nothing
6	Grotesque
7	Perky Fortune
8	Puny Officer
9	Sentinels
10	St. George Assistance Company Ltd.

artista
Autotone
Elvis Juice
Enraged Miss
Equatorial Maximum
Equatorial Nothing
Grotesque
Perky Fortune
Puny Officer
Sentinels
St. George Assistance Company Ltd.

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

```
SELECT ROUND(AVG(prezzo)::numeric,2) AS prezzo_medio
FROM biglietto, acquisto_biglietto, acquisto
WHERE biglietto.id = acquisto_biglietto.biglietto
AND acquisto_biglietto.acquisto = acquisto.id
AND biglietto.festival = 'Fog Festival 2020';
```

	prezzo_medio numeric
1	345.71

```
Inserisci scelta (da 1 a 5, -1 per uscire)3
Seleziona un edizione del festival
1 - Fog Festival 2019
2 - Fog Festival 2020
3 - Fog Festival 2022
1

+-----+
| prezzo_medio |
+-----+
| 345.71      |
+-----+
```

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

```
SELECT DISTINCT artista.nome AS artista
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;
```

	artista character varying (100)
1	Ampleknot
2	Artistic Wood
3	Between the Lads and Me
4	Blues Train
5	Brubeck Sun
6	Chris and the Cool Humans
7	Classic Drunk
8	Disciples Of Oslo
9	Duddley Smith
10	Familiar Waters
11	Honorable Address

```
+-----+
| artista |
+-----+
Ampleknot
Artistic Wood
Between the Lads and Me
Blues Train
Brubeck Sun
Chris and the Cool Humans
Classic Drunk
Disciples Of Oslo
Duddley Smith
Familiar Waters
Honorable Address
Iniquitous Darkness
Original Amore
Purple Day
Realistic Distribution
SantoS
Sense
Slowly Orbit
The Bright Monkeys
The High Street Panic
The Warning Channel
+-----+
```

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;
```

	impianto_audio tipo_audio[]	impianto_luci tipo_luci[]	numero_concerti bigint
1	{grande}	{standard}	24
2	{medio}	{standard}	7
3	{arena}	{pro}	69

6

impianto_audio	impianto_luci	numero_concerti
{grande}	{standard}	24
{medio}	{standard}	7
{arena}	{pro}	69

- Artisti
che

hanno effettuato più di due concerti in tutte le edizioni del festival

```
SELECT concerto.artista, COUNT(*) AS numero_concerti
FROM concerto
GROUP BY concerto.artista
HAVING COUNT(*) > 2
```

	artista character varying (100)	numero_concerti bigint
1	SantoS	3
2	The Warning Channel	3
3	Iniquitous Darkness	3
4	Elvis Juice	3
5	Brubeck Sun	4
6	Ampleknot	3
7	Slowly Orbit	3

artista	numero_concerti
SantoS	3
The Warning Channel	3
Iniquitous Darkness	3
Elvis Juice	3
Brubeck Sun	4
Ampleknot	3
Slowly Orbit	3
Purple Day	4
St. George Assistance Company Ltd.	3
Equatorial Nothing	3
Blues Train	3
Original Amore	3

Indice

```
CREATE INDEX Acquisto_idx ON Acquisto(ID);
```

Essendo un database implementato per un ampio riutilizzo, la parte interessante è 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++

Il codice C++ incluso nel progetto si compone di un file *main.cpp*.

L'esecuzione dello stesso avviene con il comando:

```
g++ main.cpp -I dependencies/lib -lpq -o db -std=c++11
```

all'interno della stessa cartella in cui risiede.

Per poter essere eseguito, il programma richiede la configurazione apposita di alcune costanti interagenti con il server di PostgreSQL e l'utente desiderato con i suoi dati di accesso, comprendendo anche porta e DB (nell'ordine *PG_HOST*, *PG_USER*, *PG_PASSWORD*, *PG_PORT*, *PG_DB*).

La descrizione dei metodi segue:

- *main()*, che avvia la connessione alla base di dati (tramite *PGconn* e *PGconnectdb* per impostazione dei parametri, stampa un messaggio di benvenuto e permette una scelta numerica

tra la serie di interrogazioni presentate nella sezione apposita. Il risultato viene restituito per mezzo di *PGresult*, poi pulito con *pQclear* e chiudendo la connessione con *PQfinish*.

- *selectChoice()*, metodo di input della scelta da parte dell'utente
- *doQuery()*, prendendo in input una query impostata dal metodo *main()* e restituendo il risultato sotto forma di *PGresult*; se la query non restituisce problemi (*PQresultstatus*), allora viene eseguita, altrimenti si prevede un messaggio di errore (*PQerrorMessage*)
- metodi di stampa e formattazione, in particolare *printTable()* tale settare la lunghezza delle colonne (*PQnfields*), delle righe (*PQntuples*), stampando il valore della riga in formato stringa (*PQgetvalue*). Una volta separati graficamente i valori con apposito metodo, vengono visualizzate le colonne apposite (*PQfname*) con relativo valore.

L'utente può selezionare le 6 query discusse sopra, ognuna selezionabile con un numero da 1 a 6, eseguibili con risultato riportato.

É inoltre presente anche la selezione dell'anno in cui si svolge il festival, per poter ulteriormente filtrare la ricerca Per ciascuna interrogazione:

- 1) Viene mostrato il nome del festival con maggiore sponsorizzazione
- 2) Vengono mostrati gli artisti headliner che hanno avuto più presenze in media nelle giornate del festival in cui suonano
- 3) Prezzo medio pagato dai clienti paganti per edizione di un certo festival. Grazie all'uso dell'indice la query recupera il risultato più velocemente
- 4) Artisti non headliner che hanno soggiornato nel camerino VIP più grande e più frequentato
- 5) Numero di concerti effettuati per le combinazioni di tipologia di impianto audio e luci
- 6) Artisti che hanno effettuato più di due concerti in tutte le edizioni del festival