Contents

Relazione Tecnica – Fase 1: Progettazione del Database

EventsMaster
Studente: Bosco Mattia
Classe: 5°C-IT
Data Creazione:10/12/2025

Introduzione

Il progetto EventsMaster nasce con l'obiettivo di realizzare una web application dedicata alla gestione completa della compravendita di biglietti per eventi. L'applicazione deve permettere di organizzare eventi strutturati, distribuirli nel tempo e nello spazio, associarvi intrattenitori, organizzatori e location, e infine consentire agli utenti di acquistare uno o più biglietti scegliendo settore e tipologia, con un prezzo che varia dinamicamente in base a diversi fattori. Il cuore del progetto è il database, che deve essere progettato in modo solido, coerente e facilmente estendibile. In questa prima fase si affronta quindi la progettazione concettuale e logica del database, partendo dall'analisi dei requisiti fino alla definizione dello schema ER e dello schema relazionale. A progettazione conclusa si passa alla scrittura delle istruzioni DDL per la creazione delle tabelle e delle relative chiavi, seguita dal DML per il popolamento del database tramite un dump di dati. L'attenzione è posta non solo sulla correttezza formale dello schema, ma anche sulla sua aderenza a uno scenario reale di utilizzo e sulla capacità di gestire relazioni complesse senza ridondanze inutili.

Analisi dei Requisiti

L'applicazione EventsMaster è pensata per gestire eventi pubblici di varia natura, come concerti, spettacoli o manifestazioni articolate in più appuntamenti. Dal punto di vista funzionale, il sistema deve consentire la definizione di manifestazioni generali, intese come contenitori logici di più eventi, ognuno dei quali si svolge in una data specifica, in una determinata fascia oraria e presso una precisa location. Ogni evento deve avere un prezzo base, che non rappresenta il costo finale del biglietto ma il punto di partenza per il calcolo del prezzo effettivo.

Gli eventi possono prevedere la partecipazione di uno o più intrattenitori, come artisti singoli o gruppi, che si esibiscono in specifiche fasce orarie. È quindi necessario gestire il tempo come entità separata, in modo da evitare duplicazioni di esibizioni identiche su più eventi o orari. Allo stesso modo, un evento può coinvolgere più organizzatori, ciascuno con un ruolo ben definito, rendendo indispensabile una relazione molti-a-molti tra eventi e organizzatori.

La gestione delle location è un altro aspetto centrale: ogni evento si svolge in una sola location, caratterizzata da un nome e da un indirizzo strutturato. Ogni location è suddivisa in settori, ciascuno con un numero limitato di posti

e un moltiplicatore di prezzo che influisce sul costo finale del biglietto. Questo consente di modellare situazioni reali come platea, tribuna o VIP, mantenendo flessibilità nel calcolo dei prezzi.

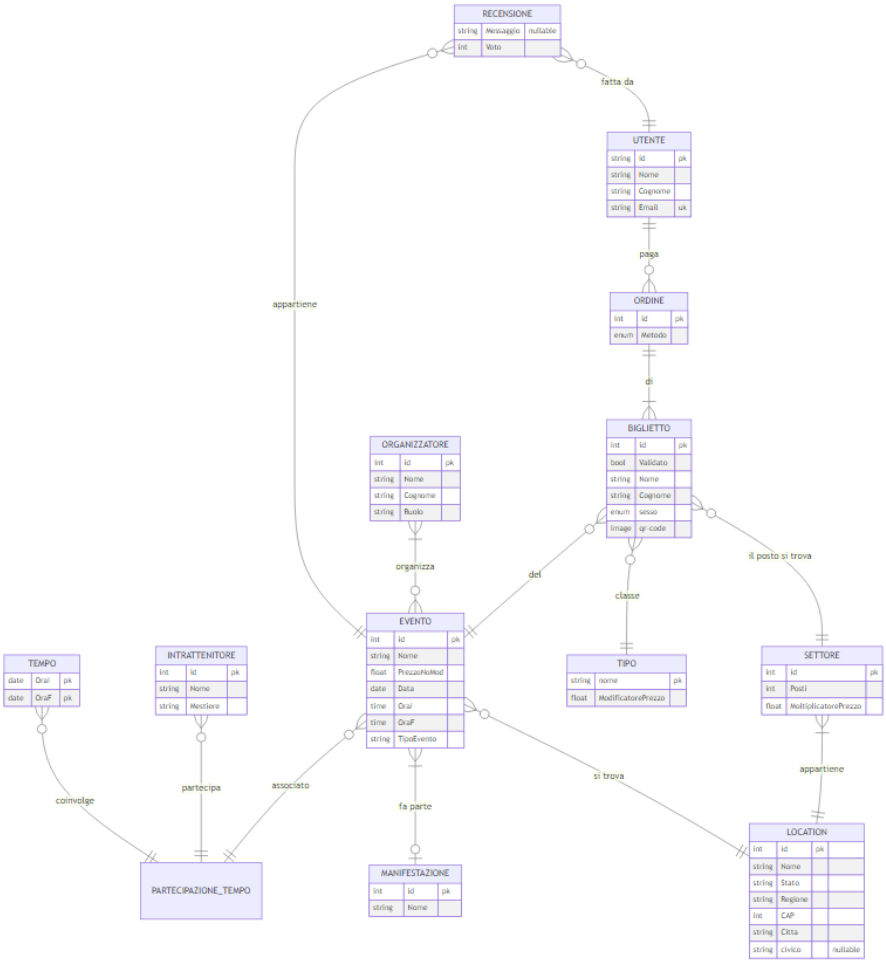
Dal punto di vista dell'utente finale, il sistema deve permettere l'acquisto di uno o più biglietti tramite un ordine, specificando il metodo di pagamento utilizzato. Ogni biglietto è associato a un singolo evento, a una tipologia (standard, ridotto, VIP, ecc.) e a un settore, e contiene le informazioni della persona che lo utilizzerà. Il biglietto deve inoltre essere identificabile in modo univoco e rappresentabile digitalmente tramite un QR-code, che consente la validazione all'ingresso dell'evento.

Gli utenti registrati nel sistema possono inoltre lasciare recensioni sugli eventi a cui hanno partecipato. Ogni recensione è legata a un solo evento e a un solo utente e prevede obbligatoriamente un voto, mentre il commento testuale è facoltativo. Questo vincolo evita recensioni duplicate e garantisce coerenza nei dati.

Dal punto di vista dei requisiti non funzionali, il database deve garantire integrità referenziale, evitando dati orfani o incoerenti. Le chiavi primarie e le chiavi esterne devono essere definite in modo esplicito, così come i vincoli di unicità e di nullabilità. La struttura deve essere normalizzata per ridurre al minimo la ridondanza, ma senza compromettere la leggibilità dello schema o la semplicità delle interrogazioni. Infine, il modello deve essere pensato per poter essere esteso in futuro, ad esempio aggiungendo nuovi metodi di pagamento, nuove tipologie di biglietto o funzionalità avanzate di analisi delle vendite.

Il Database

Diagramma ER



Lo schema relazionale

MANIFESTAZIONI

- id **PK**
- Nome

La tabella MANIFESTAZIONI rappresenta un contenitore logico di eventi accomunati da un tema o da un contesto comune. Ogni manifestazione è identificata da un codice univoco ed è descritta da un nome significativo. Non contiene informazioni temporali o spaziali, che vengono invece delegate agli eventi che ne fanno parte.

MANIFESTAZIONE_EVENTI

- *idManifestazione* (Manifestazione–>id)
- *idEvento* (Evento–>id)
- **PK(idManifestazione, idEvento)**

La tabella MANIFESTAZIONE_EVENTI modella l'associazione tra una manifestazione e gli eventi che la compongono. È una tabella di collegamento necessaria per rappresentare il programma di una manifestazione e consente di gestire correttamente il caso in cui una manifestazione includa più eventi distinti.

EVENTI

- id **PK**
- *idManifestazione* (Manifestazione–>id)
- *idlocation* (Location–>id)
- Nome
- PrezzoNoMod
- Data
- OraI
- OraF

La tabella EVENTI rappresenta l'unità fondamentale del sistema. Ogni evento è identificato da un codice univoco ed è associato a una manifestazione e a una location. Oltre al nome, vengono memorizzate la data e la fascia oraria di svolgimento, necessarie per la costruzione del programma. Il prezzo memorizzato è il prezzo base, che verrà successivamente modificato in base al settore e alla tipologia di biglietto scelta.

INTRATTENITORI

- id **PK**
- Nome
- Mestiere

La tabella INTRATTENITORI memorizza gli artisti o i gruppi che partecipano agli eventi. L'attributo Nome rappresenta il nome d'arte o del gruppo, mentre Mestiere identifica la tipologia di intrattenitore, come cantante, comico o altro. La separazione degli intrattenitori dagli eventi consente il riutilizzo degli stessi in più contesti.

TEMPI

- OraI **PK**
- OraF **PK**

La tabella TEMPI rappresenta una fascia oraria definita da un'ora di inizio e un'ora di fine. Questa entità consente di evitare la duplicazione delle stesse fasce orarie per esibizioni diverse e rende più flessibile la gestione del programma degli eventi.

ESIBIZIONI

- *idIntrattenitore* (Intrattenitore–>id)
- *idEvento* (Evento–>id)
- *OraI* (Tempo–>OraI)
- *OraF* (Tempo–>OraF)
- **PK(idEvento, idIntrattenitore, OraI, OraF)**

La tabella ESIBIZIONI rappresenta la relazione tra un evento, un intrattenitore e una specifica fascia oraria. Ogni esibizione identifica quindi chi si esibisce, in quale evento e in quale intervallo di tempo, permettendo di descrivere programmi complessi con più artisti e più momenti all'interno dello stesso evento.

RECENSIONI

- *idEvento* (Evento–>id)
- *idUtente* (Utente–>id)
- Voto
- Messaggio ***NULL***
- **PK(idEvento, IdUtente)**

La tabella RECENSIONI memorizza le valutazioni lasciate dagli utenti sugli eventi. Ogni recensione è univocamente identificata dall'evento recensito e dall'utente che l'ha scritta, impedendo recensioni duplicate. Il voto è obbligatorio, mentre il messaggio testuale è facoltativo, permettendo sia valutazioni rapide sia commenti più dettagliati.

ORGANIZZATORI

- id **PK**
- Nome
- Cognome
- Ruolo

L'organizzatore è caratterizzato da nome e cognome più il ruolo che ricopre nell'organizzazione dell'evento.

ORGANIZZATORI_EVENTO

- *idEvento* (Evento–>id)
- *idOrganizzatore* (Organizzatore–>id)
- **PK(idEvento, idOrganizzatore)**

La tabella ORGANIZZATORI rappresenta le persone coinvolte nell'organizzazione degli eventi. Oltre ai dati anagrafici di base, viene memorizzato il ruolo ricoperto, utile per distinguere responsabilità e funzioni all'interno dell'organizzazione.

LOCATIONS

- id **PK**
- Nome
- Indirizzo
  - Stato
  - Regione
  - CAP
  - Città
  - civico ***NULL***

La tabella LOCATIONS rappresenta i luoghi fisici in cui si svolgono gli eventi. Ogni location è identificata da un codice univoco ed è descritta tramite un nome e un indirizzo strutturato. Il numero civico è opzionale, poiché non sempre disponibile o significativo.

SETTORI

- id **PK**
- *idLocation* (Location–>id)
- Posti
- MoltiplicatorePrezzo

La tabella SETTORI rappresenta le suddivisioni interne di una location. Ogni settore è associato a una specifica location e contiene il numero di posti disponibili e un moltiplicatore di prezzo, utilizzato per il calcolo del costo finale del biglietto in base alla posizione.

BIGLIETTI

- id **PK**
- *idEvento* (Evento–>id)
- *idClasse* (Tipo–>id)
- Check
- Nome
- Cognome
- Sesso
- QR-code

La tabella BIGLIETTI rappresenta il titolo di accesso a un evento. Ogni biglietto è associato a un evento e a una tipologia di biglietto e contiene i dati della persona che ne usufruisce. Il QR-code consente la rappresentazione digitale del biglietto e la sua validazione, mentre il campo di controllo indica se il biglietto è già stato utilizzato.

SETTORE_BIGLIETTI

- *idSettore* (Settore–>id)
- *idBiglietto* (Biglietto–>id)
- Posto
  - Fila
  - Numero
- **PK(idSettore, idBiglietto)**

La tabella SETTORE_BIGLIETTI associa un biglietto a un settore specifico e definisce il posto assegnato tramite fila e numero. Questa struttura permette di gestire posti numerati e garantisce che ogni biglietto sia collegato a una posizione precisa all'interno della location.

ORDINI

- id **PK**
- Metodo

La tabella ORDINI rappresenta una transazione di acquisto effettuata da un utente. Ogni ordine è identificato da un codice univoco e memorizza il metodo di pagamento utilizzato. Un ordine può comprendere uno o più biglietti.

ORDINE_BIGLIETTI

- *idOrdine* (Ordine–>id)
- *idBiglietto* (Biglietto–>id)
- **PK(idOrdine, idBiglietto)**

La tabella ORDINE_BIGLIETTI è una tabella di collegamento che associa i biglietti a un ordine. Consente di gestire acquisti multipli all'interno della stessa transazione, mantenendo separata la logica dell'ordine da quella del singolo biglietto.

UTENTE

- id **PK**
- Nome
- Cognome
- Email **UK**

La tabella UTENTE rappresenta l'utente registrato al sistema. L'utente può effettuare ordini e scrivere recensioni. Oltre ai dati anagrafici di base, viene memorizzata un'email univoca, utilizzabile sia per l'autenticazione sia per comunicazioni come newsletter o notifiche.

UTENTE_ORDINI

- *idOrdine* (Ordine–>id)
- *idUtente* (Utente–>id)
- **PK(idOrdine, idUtente)**

La tabella UTENTE_ORDINI collega gli utenti agli ordini effettuati. Questa struttura consente di gestire correttamente la relazione tra utenti e ordini e mantiene il modello flessibile in caso di future estensioni del sistema.

TIPO

- nome **PK**,
- ModificatorePrezzo

La tabella TIPO rappresenta la tipologia del biglietto, come standard, ridotto o VIP. Ogni tipo è caratterizzato da un modificatore di prezzo che viene applicato al prezzo base dell'evento, permettendo una gestione dinamica e scalabile dei costi.