**Elaborato per il corso di Basi di dati**

**A.A 2022/2023**

Progetto di una base di dati per la gestione della stagione concertistica di un’orchestra

De Nardi Chiara

[chiara.denardi2@studio.unibo.it](mailto:chiara.denardi2@studio.unibo.it)

0001027282



Università degli studi di Bologna

Campus di Cesena

Facoltà di Ingegneria e Scienze Informatiche

**INDICE**

ANALISI DEI REQUISITI 3

ESTRAZIONE DEI CONCETTI PRINCIPALI 3

INTERVISTA 3

PROGETTAZIONE CONCETTUALE 5

SCHEMA SCHELETRO 5

PROGETTAZIONE LOGICA 6

STIMA DEL VOLUME DEI DATI 6

DESCRIZIONE DELLE OPERAZIONI PRINCIPALI E STIMA DELLA LORO FREQUENZA……………………………………..6

SCHEMI DI NAVIGAZIONE E TABELLE DEGLI ACCESSI…………………………………………………………………………………8

RAFFINAMENTO DELLO SCHEMA…………………………………………………………………………………………………………….10

ANALISI DELLE RIDONDANZE…………………………………………………………………………………………………………………..10

TRADUZIONE DI ENTITA’ E ASSOCIAZIONI IN RELAZIONE…………………………………………………………………………11

SCHEMA RELAZIONALE FINALE………………………………………………………………………………………………………………..12

TRADUZIONE DELLE OPERAZIONI IN QUERY SQL……………………………………………………………………………………..15

PROGETTAZIONE DELL’APPLICAZIONE…………………………………………………………………………………………………………17

DESCRIZIONE DELL’ARCHITETTURA DELL’APPLICAZIONE REALIZZATTA……………………………………………………..17

**ANALISI DEI REQUISITI**

Si richiede di organizzare un sistema informativo per la gestione della stagione concertistica dell’anno corrente di un'orchestra. La base di dati dovrà mantenere traccia delle informazioni relative ai concerti fatti e dei relativi orchestrali, dei brani suonati e dei biglietti venduti.

**INTERVISTA**

Un primo testo ottenuto dall’intervista è il seguente:

Si vuole tenere traccia della stagione concertistica dell’anno corrente di un’orchestra. Durante l’anno l’orchestra si sposterà in diverse città (di cui si memorizza nome e provincia) e si esibirà in una struttura, come un teatro o un palazzetto. Di questa struttura si tiene presente il nome, l’indirizzo, la capienza. L’orchestra però può esibirsi più volte in una stessa città ma in strutture diverse e durante un concerto si esibisce solo un’orchestra. Un’orchestra non fa più di un concerto al giorno ed essa è caratterizzata da un nome, un direttore musicale (ciascuno con nome e cognome) che può anche cambiare in occasioni speciali, un insieme di orchestrali (con matricola, nome, cognome e uno o più strumenti suonati) e vari ospiti speciali (tipo violinisti o pianisti che vengono invitati). Ogni concerto viene tenuto in una certa data e orario ed ha un titolo, una descrizione e una scaletta con i pezzi musicali che verranno suonati. Anche ogni brano ha un codice, un titolo e può essere stato composto da uno o più artisti (ciascuno con codice e nome). Uno stesso pezzo può essere rappresentato in diversi concerti dell’orchestra. Viene tenuta traccia anche dei biglietti (sezione zona, costo, codice) e di eventuali sconti per studenti o riduzioni varie e anche delle persone che hanno comprato il biglietto per un determinato concerto, memorizzandone nome, cognome, codice fiscale, numero di telefono e una mail. Viene infine segnalato anche quando un concerto è sold out e quindi non verranno venduti ulteriori biglietti.

**ESTRAZIONE DEI CONCETTI PRINCIPALI**

A seguito della lettura e comprensione dei requisiti richiesti, si procede sviluppando un testo che ne riassuma tutti i concetti e in particolare ne estragga quelli principali, risultando essere in questo modo meglio fruibile per la realizzazione della base di dati. Si tiene conto delle seguenti correzioni di ambiguità.

|  |  |
| --- | --- |
| **Termine usato** | **Sinonimo** |
| orchestrali | musicisti |
| struttura | edificio |
| Ospiti speciali | musicisti |
| pezzo | Brano musicale |
| Persone che hanno comprato il biglietto | clienti |

Si vuole tenere traccia della stagione concertistica dell’anno corrente di un’orchestra. Durante l’anno l’orchestra si sposterà in diverse città e si esibirà in un edificio designato. Di quest’ultimo si tiene presente il nome, l’indirizzo e la capienza. L’orchestra però può esibirsi più volte in una stessa città ma mai nello stesso edificio e durante un concerto si esibisce solo un’orchestra. Un’orchestra non fa più di un concerto al giorno ed essa è caratterizzata da un nome, un direttore musicale che può anche essere cambiato in occasioni speciali, un insieme di musicisti identificati da un codice univoco e che suonano uno o più strumenti musicali e infine può accadere che presenzino anche ulteriori musicisti più famosi come ospiti della serata. Ogni concerto viene tenuto in una certa data e orario ed ha titolo, una descrizione e una scaletta con i brani musicali che verranno suonati. Anche ogni brano ha un codice, un titolo e può essere stato composto da uno o più artisti, inoltre può essere rappresentato in diversi concerti dell’orchestra in date diverse.

Viene tenuta traccia anche dei biglietti (sezione zona, costo, codice) e di eventuali sconti per studenti o riduzioni varie e anche dei clienti che ci saranno in un determinato concerto, memorizzandone nome, cognome, codice fiscale, numero di telefono e una mail. Viene infine segnalato anche quando un concerto è sold out e quindi non verranno venduti ulteriori biglietti.

Elenco delle principali azioni richieste:

1. Aggiungere una nuova data alla stagione concertistica con relative informazioni
2. Registrare l’acquisto di un nuovo biglietto per una data del concerto da parte di un cliente
3. Cercare la città e il giorno dei concerti che si terranno in una certa provincia, esplicitando anche nome e cognome degli orchestrali
4. Mostrare tutte le serate in cui c’è sold out, specificando il numero dei biglietti venduti
5. Seleziona i musicisti presenti al concerto in una certa orchestra
6. Mostrare l’elenco della scaletta dei brani di un concerto in cui è presente un certo direttore musicale
7. Mostrare a che concerto può andare una persona con un certo biglietto
8. Aumentare il prezzo di tutti i biglietti di 5 euro

**PROGETTAZIONE CONCETTUALE**

**SCHEMA SCHELETRO**

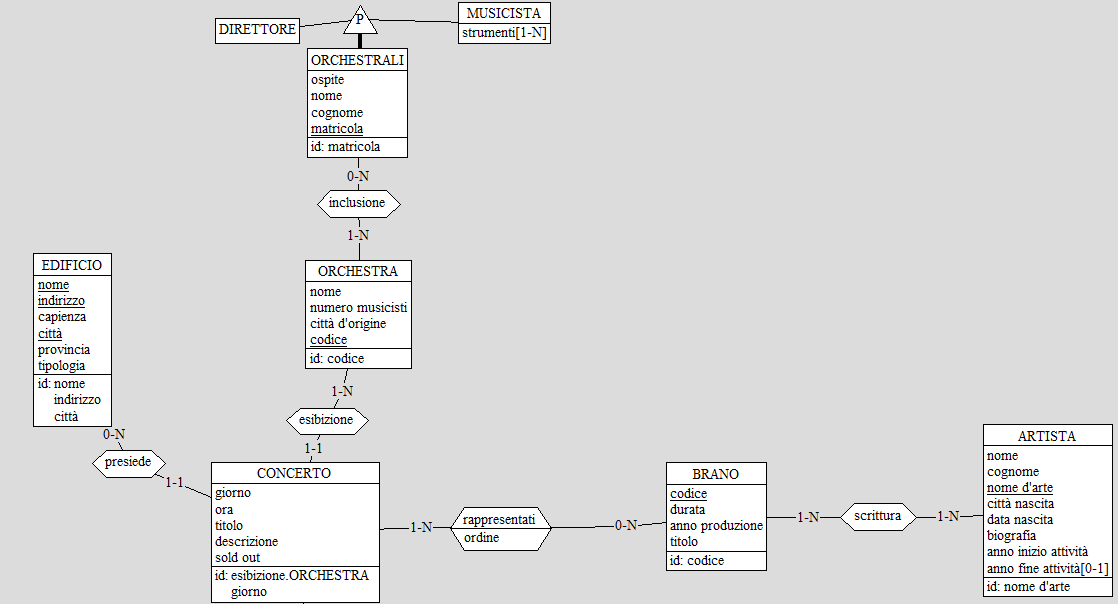


Figura : Schema E/R con le principali entità per la modellazione dei concerti e della scaletta dei brani

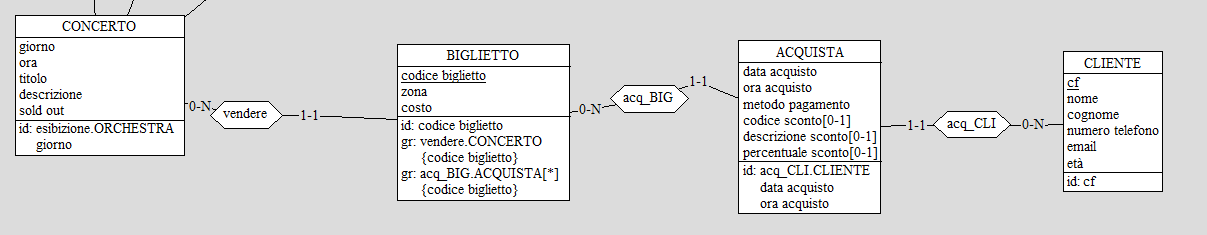


Figura : Schema E/R rappresentante il sistema dei biglietti, degli sconti e degli spettatori

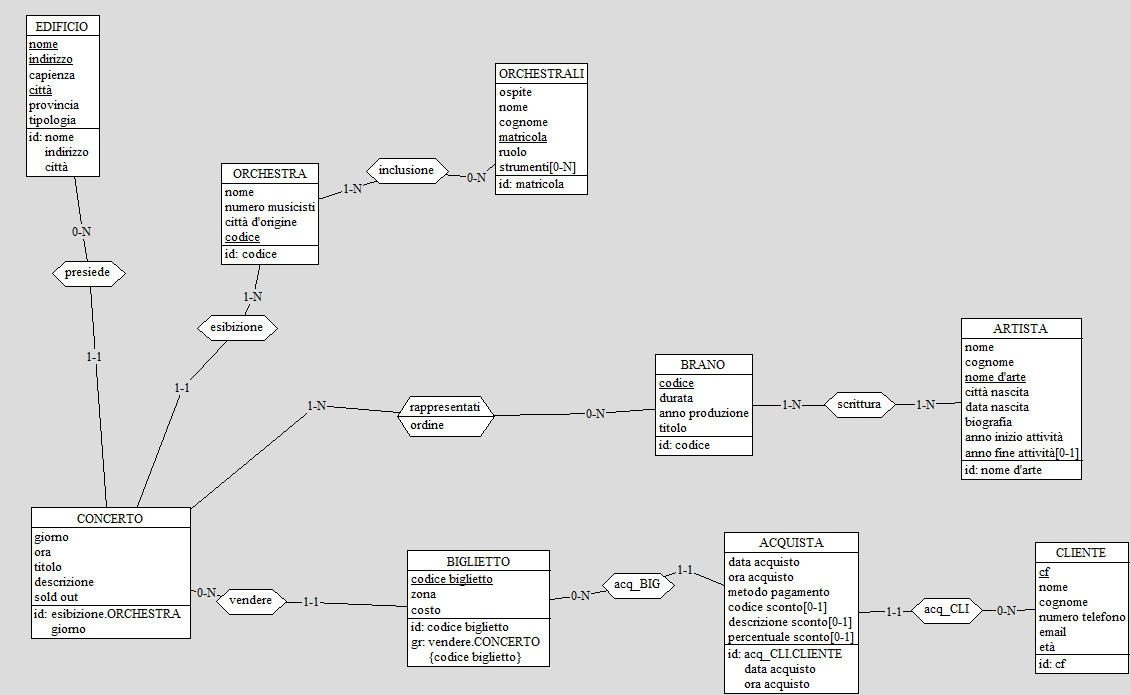


Figura : schema E/R finale

**PROGETTAZIONE LOGICA**

**STIMA DEL VOLUME DEI DATI**

Si fornisce in questa fase una tabella contenente il numero medio di istanze di ogni entità e associazione dello schema globale: i valori di carico fanno riferimento alle partecipazioni medie riportate in figura 3.

|  |  |  |
| --- | --- | --- |
| **CONCETTO** | **COSTRUTTO** | **VOLUME** |
| BIGLIETTO | E | 20000 |
| ACQUISTA | E | 20000 |
| CLIENTE | E | 20000 |
| CONCERTO | E | 50 |
| ORCHESTRA | E | 50 |
| *inclusione* | R | 6000 |
| ORCHESTRALI | E | 4000 (circa 80 per orchestra) |
| DIRETTORE | E | 60 |
| MUSICISTA | E | 3940 |
| EDIFICIO | E | 50 |
| *rappresentanti* | R | 6000 |
| BRANO | E | 2000 (circa 40 per concerto) |
| *scrittura* | R | 8000 |
| ARTISTA | E | 6000 (1 brano è scritto in media da 3 persone) |

**DESCRIZIONE DELLE OPERAZIONI PRINCIPALI E STIMA DELLA LORO FREQUENZA**

Le operazioni da effettuare sono quelle già elencate nella fase di analisi. Segue una tabella riportante la loro descrizione e relativa frequenza:

|  |  |  |  |
| --- | --- | --- | --- |
| **CODICE** | **OPERAZIONE** | **FREQUENZA** | **TIPO (interattiva/batch)** |
| 1 | Aggiungere una nuova data alla stagione concertistica con relative informazioni | 4 al mese | I |
| 2 | Registrare l’acquisto di un nuovo biglietto per una data del concerto da parte di un cliente | 80 a settimana | I |
| 3 | Cercare la città e il giorno dei concerti che si terranno in una certa provincia, esplicitando anche nome e cognome degli orchestrali | 60 a settimana | I |
| 4 | Mostrare tutte le serate in cui c’è sold out, specificando il numero dei biglietti venduti | 3 al mese | I |
| 5 | Seleziona i musicisti presenti al concerto in una certa orchestra | 80 a settimana | I |
| 6 | Mostrare l’elenco della scaletta dei brani di un concerto in cui è presente un certo direttore musicale | 20 a settimana | I |
| 7 | Mostrare a che concerto può andare una persona con un certo biglietto | 40 al mese | I |
| 8 | Aumentare il prezzo di tutti i biglietti di 5 euro | 1 all’anno | B |

**SCHEMI DI NAVIGAZIONE E TABELLE DEGLI ACCESSI**

Sono riportate in seguito le tabelle degli accessi delle operazioni sopra riportate; inoltre, ove non risulti banale, sono stati inseriti i relativi schemi di navigazione. Al fine del calcolo dei costi, si considerano di peso doppio gli accessi in scrittura rispetto a quelli in lettura.

**OP 1 - Aggiungere una nuova data alla stagione concertistica** **con relative informazioni**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Concerto | E | 1 | S |

TOT= 1 S x2 = 2 x 4 mese= 8 al mese

**OP 2 – Registrare l’acquisto di un nuovo biglietto per una data del concerto da parte di un cliente**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| cliente | E | 1 | L |
| Cliente | E | 1 | S |
| Acq\_cli | R | 2 | S |
| Acquista | E | 2 | S |
| Acq\_big | R | 1 | S |
| Biglietto | E | 1 | S |

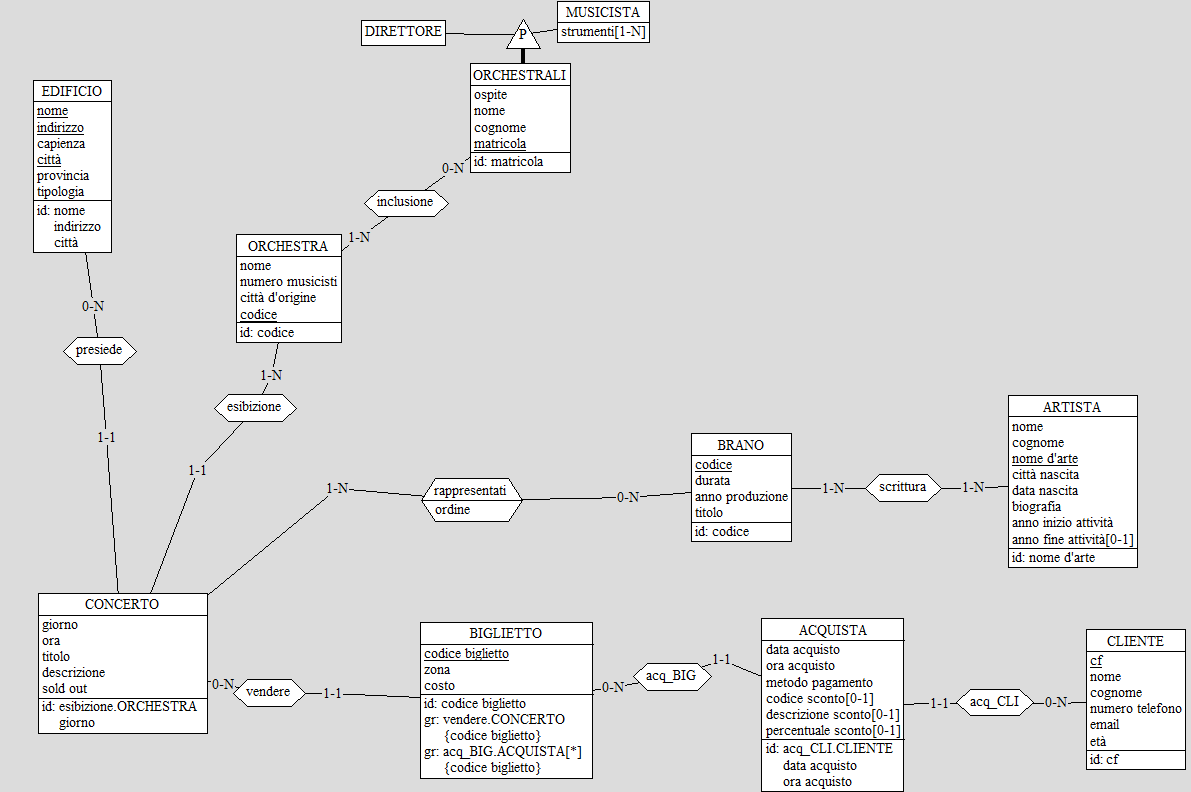
TOT=7s + 1L🡪15x80 a sett= 1200 a settimana

**OP 3 - Cercare la città e il giorno dei concerti che si terranno in una certa provincia, esplicitando anche nome e cognome degli orchestrali**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Edificio | E | 1 | L |
| Presiede | R | 1 | L |
| Concerto | E | 1 | L |
| Esibizione | R | 1 | L |
| Orchestra | E | 1 | L |
| Inclusione | R | 1 | L |
| Orchestrali | E | 1 | L |
| Direttore | E | 1 | L |

TOT=8L 🡪8x60 a settimana = 480 a settimana

**OP 4 - Mostrare tutte le serate in cui c’è sold out, specificando il numero dei biglietti venduti**



|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Concerto | E | 1 | L |
| Presiede | R | 1 | L |
| Edificio | E | 1 | L |
| Vendere | R | 1 | L |
| Biglietti | E | 1 | L |
| Acqu\_big | R | 1 | L |
| Acquista | E | 1 | L |

TOT=7L 🡪 7 x 3 al mese = 21 al mese

**OP 5 – seleziona i musicisti presenti al concerto in una certa orchestra**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Concerto | E | 1 | L |
| Esibizione | R | 1 | L |
| Orchestra | E | 1 | L |
| Inclusione | R | 120 | S |
| Orchestrali | E | 120 | S |
| Musicista | E | 120 | S |

TOT=360S + 3L 🡪 720+2 x 80 a sett= 57760 a settimana

**OP 6 - Mostrare l’elenco della scaletta dei brani di un concerto in cui è presente un certo direttore musicale**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Direttore | E | 1 | L |
| Orchestrali | E | 1 | L |
| Inclusione | R | 1 | L |
| Orchestra | E | 1 | L |
| Esibizione | R | 1 | L |
| Concerto | E | 1 | L |
| Rappresentanti | R | 40 | L |
| Brano | E | 40 | L |
| Scrittura | R | 40 | L |
| Artista | E | 40 | L |

TOT=166L 🡪 166x20 a sett = 3320 a settimana

**OP 7 - Mostrare a che concerto può andare una persona con un certo biglietto**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Biglietto | E | 1 | L |
| Vendere | R | 1 | L |
| Concerto | E | 1 | L |

TOT=3L 🡪 3x40 al mese = 120 al mese

**OP 8 - Aumentare il prezzo di tutti i biglietti di 5 euro**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO |
| Concerto | E | 50 | S |
| Vendere | R | 50 | S |
| Biglietto | E | 50 | S |

TOT=150S = 150x1 all’anno = 150 all’anno

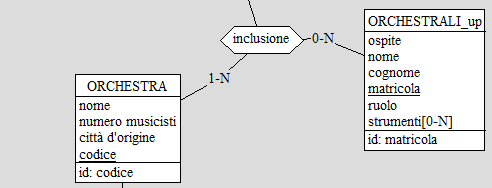
**ANALISI DELLE RIDONDANZE**

Nello schema non sono presenti attributi ridondanti con le richieste presentate. Quindi non verranno valutate le ridondanze.

**RAFFINAMENTO DELLO SCHEMA**

**Eliminazione delle gerarchie**

Per la gerarchia Orchestrali (t, e), si decide di adottare come soluzione il collasso verso l’alto. La scelta è motivata dal fatto che nonostante debbano essere aggiunti 2 attributi (ruolo e strumenti) alla tabella Orchestrali è comunque preferibile avere un’unica tabella che contiene tutti gli orchestrali piuttosto che due tabelle divise che non avrebbero alcuna utilità, poiché gli accessi all’entità padre e alle entità figlie sono contestuali.



**Eliminazione degli attributi composti**

È possibile individuare nello schema ER che l’attributo “Indirizzo” dell’entità Edificio è un attributo composto dai sotto-attributi: “Via”, “nCivico”, “Cap”. Esso viene quindi scomposto nelle sue tre sotto-parti per entrambe le entità in questione.

**Scelta delle chiavi primarie**

Nello schema sono già evidenziate senza ambiguità tutte le chiavi primarie per la maggior parte delle entità. Nell’entità Edificio, dato che la chiave è composta da 3 primarie la posso trasformare in un’unica chiave chiamata IdEdificio.

**Eliminazione degli identificatori esterni**

Nello schema E/R sono eliminate le seguenti relazioni:

* Inclusione, reificata importando matricolaOrchestrali da Orchestrali e codiceOrchestra da Orchestra
* Esibizione, importando Codice come CodiceOrchestra in Concerto
* Presiede, importando IdEdificio in Concerto
* Rappresentati, reificata importando codiceBrano da Brano, giornoConcerto da Concerto e codiceOrchestra da Orchestra
* Vendere, importando IdConcerto in Biglietto
* Acq\_big, importando IdBiglietto in Acquista
* Acq\_cli, importando cf come cfCliente in Acquista
* Scrittura, reificata importando codiceBrano da Brano e nomeArtista da Artista

**TRADUZIONE DELLE ENTITÀ E ASSOCIAZIONI IN RELAZIONI**

EDIFICIO (nome, via, nCivico, cap, capienza, città, provincia, tipologia)

CONCERTO (giorno, ora, titolo, descrizione, sold out, CodiceOrchestra: Orchestra, idEdificio: Edificio)

ORCHESTRA (nome, numero musicisti, città d’origine, codice)

ORCHESTRALI (ospite, nome, cognome, matricola, ruolo)

BRANO (codice, durata, anno produzione, titolo)

ARTISTA (nome, cognome, nomeArtista, città nascita, data nascita, biografia, anno inizio attività, anno fine attività\*)

BIGLIETTO (codice biglietto, zona, costo, giornoConcerto: Concerto, codiceOrchestra: Orchestra)

ACQUISTA (data acquisto, ora acquisto, metodo pagamento, codice sconto\*, descrizione sconto\*, percentuale sconto\*, cf: Cliente, codiceBiglietto: Biglietto)

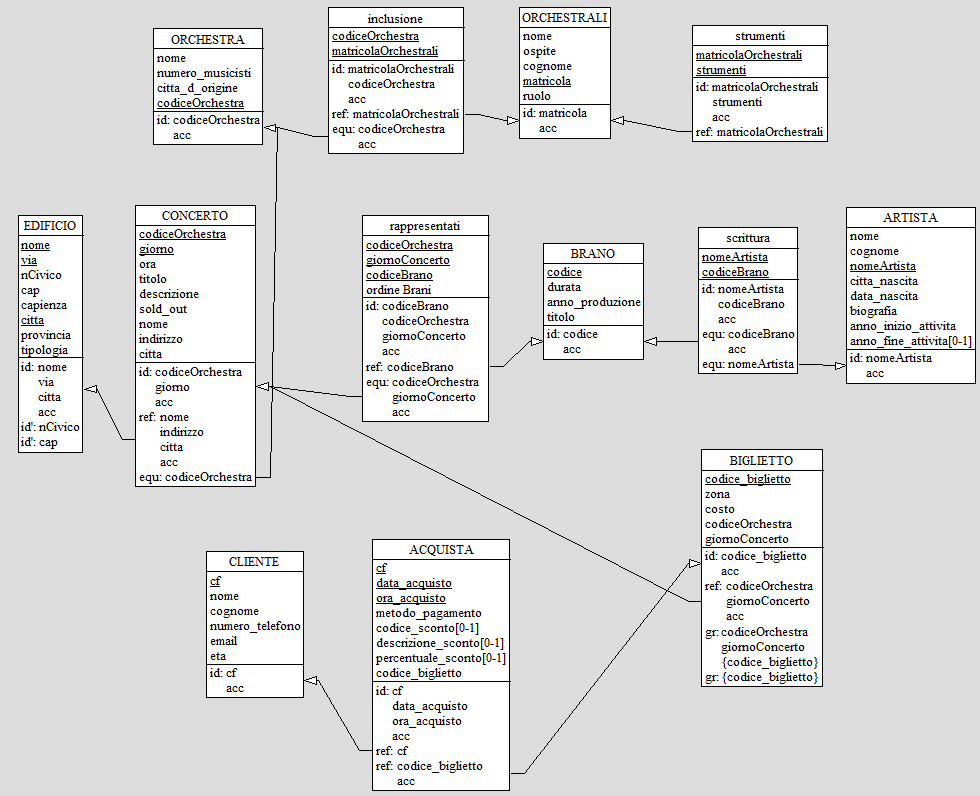
CLIENTE (cf, nome, cognome, numero telefono, email, età)

INCLUSIONE (matricolaOrchestrali: Orchestrali, codiceOrchestra: Orchestra)

SCRITTURA (codiceBrano: Brano, nomeArtista: Artista)

RAPPRESENTANTI (codiceBrano: Brano, codiceOrchestra: Orchestra, giornoConcerto: Concerto, ordineBrani)

**SCHEMA RELAZIONALE FINALE**



**COSTRUZIONE DELLE TABELLE DEL DB IN LINGUAGGIO SQL**

* ORCHESTRA
  + CREATE TABLE Orchestra (

Nome VARCHAR (20) NOT NULL,

Numero\_musicisti INT (50) NOT NULL,

Città\_origine VARCHAR (20) NOT NULL,

CodiceOrchestra INT PRIMARY KEY

)

* INCLUSIONE
  + CREATE TABLE Inclusione (

CodiceOrchestra INT NOT NULL FOREIGN KEY REFERENCES Orchestra

ON DELETE CASCADE

ON UPDATE NO ACTION,

matricolaOrchestrali INT NOT NULL FOREIGN KEY REFERENCES Orchestrali

ON DELETE CASCADE

ON UPDATE NO ACTION,

PRIMARY KEY (codiceOrchestra, matricolaOrchestrali)

)

* ORCHESTRALI
  + CREATE TABLE Orchestrali (  
    Nome VARCHAR (20) NOT NULL,

Ospite VARCHAR (2) NOT NULL CHECK (Ospite IN (‘sì’, ‘no’)),

Cognome VARCHAR (20) NOT NULL

Matricola INT PRIMARY KEY

Ruolo VARCHAR (9) NOT NULL CHECK (Ospite IN (‘musicista’, ‘direttore’)),

)

* STRUMENTI
  + CREATE TABLE Strumenti (  
    matricolaOrchestrali INT NOT NULL FOREIGN KEY REFERENCES Orchestrali

ON DELETE CASCADE

ON UPDATE NO ACTION

Strumenti VARCHAR (20) NOT NULL CHECK (Strumenti IN (‘violini’, ‘viole’, ‘violoncello’, ‘contrabbasso’, ‘arpe’, ‘flauto’, ‘corni’, ‘clarinetti’, ‘percussioni’, ‘trombe’, ‘oboi’, ‘fagotti’, ‘ottavini’)

PRIMARY KEY (strumenti, matricolaOrchestrali)  
)

* EDIFICIO
  + CREATE TABLE Edificio (  
    nome VARCHAR (20) NOT NULL,

Via VARCHAR (30) NOT NULL,

nCivico INT NOT NULL,

Cap INT NOT NULL CHECK (LEN (cap)=5),

Capienza INT NOT NULL,

Città VARCHAR (20) NOT NULL,

Provincia VARCHAR (20) NOT NULL,

Tipologia VARCHAR (20) NOT NULL CHECK (Tipologia IN (‘teatro’, ‘palazzetto’)),

PRIMARY KEY (nome, via, nCivico, cap, città)  
)

* CONCERTO (  
  Giorno DATE NOT NULL,

Ora DATETIME NOT NULL,

Titolo VARCHAR (20) NOT NULL,

Descrizione VARCHAR (50) NOT NULL,

Sold out VARCHAR (2) NOT NULL CHECK (Ospite IN (‘sì’, ‘no’)),

NomeEdificio VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES Edificio  
 ON DELETE CASCADE

ON UPDATE NO ACTION,

codiceOrchestra INT NOT NULL FOREIGN KEY REFERENCES Orchestra

ON DELETE CASCADE

ON UPDATE NO ACTION,

Via VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES Edificio

ON DELETE CASCADE

ON UPDATE NO ACTION,

nCivico INT NOT NULL FOREIGN KEY REFERENCES Edificio

ON DELETE CASCADE

ON UPDATE NO ACTION,

Cap INT NOT NULL CHECK (LEN (cap)=5) FOREIGN KEY REFERENCES Edificio

ON DELETE CASCADE

ON UPDATE NO ACTION,

Città VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES Edificio

ON DELETE CASCADE

ON UPDATE NO ACTION,

PRIMARY KEY (giorno, nomeEdificio, via, nCivico, cap, codiceOrchestra)  
)

* RAPPRESENTANTI
  + CREATE TABLE Rappresentanti (  
    ordineBrani INT NOT NULL,

CodiceBrano INT NOT NULL FOREIGN KEY REFERENCES Brano

ON DELETE CASCADE

ON UPDATE NO ACTION,

CodiceOrchestra INT NOT NULL FOREIGN KEY REFERENCES Orchestra

ON DELETE CASCADE

ON UPDATE NO ACTION,

giornoConcerto DATE NOT NULL FOREIGN KEY REFERENCES Concerto

ON DELETE CASCADE

ON UPDATE NO ACCTION,

PRIMARY KEY (ordineBrani, codiceBrano, CodiceOrchestra, giornoConcerto)  
)

* BRANO
  + CREATE TABLE Brano (  
    codiceBrano INT PRIMARY KEY,

Durata TIME NOT NULL,

AnnoProd DATE NOT NULL,

Titolo VARCHAR (20) NOT NULL  
)

* SCRITTURA
  + CREATE TABLE Scrittura (

NomeArtista VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES Artista

ON DELETE CASCADE

ON UPDATE NO ACTION,

codiceBrano INT NOT NULL FOREIGN KEY REFERENCES Brano

ON DELETE CASCADE

ON UPDATE NO ACTION,

PRIMARY KEY (nomeArtista, codiceBrano)

)

* ARTISTA
  + CREATE TABLE Artista (

Nome VARCHAR (20) NOT NULL,

Cognome VARCHAR (20) NOT NULL,

NomeArtista VARCHAR (20) PRIMARY KEY,

CittàNascita VARCHAR (20) NOT NULL,

DataNascita DATE NOT NULL,

Biografia VARCHAR (50) NOT NULL,

AnnoInizioAttività DATE NOT NULL,

AnnoFineAttività DATE  
)

* CLIENTE
  + CREATE TABLE Cliente (

Cf VARCHAR (16) PRIMARY KEY,

Nome VARCHAR (20) NOT NULL,

Cognome VARCHAR (20) NOT NULL,

Età INT NOT NULL,

Email VARCHAR (30) NOT NULL,

NumTel INT (15) NOT NULL

)

* ACQUISTA
  + CREATE TABLE Acquista (

DataAcq DATA NOT NULL,

OraAcq TIME NOT NULL,

MetodoPagamento VARCHAR (20) NOT NULL,

CodiceSconto INT ,

DescrSconto VARCHAR (50),

PercSconto INT (3),

Cf VARCHAR (15) NOT NULL FOREIGN KEY REFERENCES Cliente

ON DELETE CASCADE

ON UPDATE NO ACTION,

CodiceBiglietto INT NOT NULL FOREIGN KEY REFERENCES Biglietto

ON DELETE CASCADE

ON UPDATE NO ACTION,

PRIMARY KEY (cf, codiceBiglietto, dataAcq, oraAqc)

)

* BIGLIETTO
  + CREATE TABLE Biglietto (

codiceBiglietto INT NOT NULL,

zona VARCHAR (20) NOT NULL,

costo INT NOT NULL CHECK (costo >= 0),

codiceOrchestra INT NOT NULL FOREIGN KEY REFERENCES Orchestra

ON DELETE CASCADE

ON UPDATE NO ACTION,

giornoConcerto DATE NOT NULL FOREIGN KEY REFERENCES Concerto

ON DELETE CASCADE

ON UPDATE NO ACTION,

PRIMARY KEY (codiceBiglietto, codiceOrchestra, giornoConcerto)

)

**TRADUZIONE DELLE OPERAZIONI IN QUERY SQL**

1. **Aggiungere una nuova data alla stagione concertistica con relative informazioni**

INSERT INTO Concerto (ora, giorno, titolo, descrizione, sold\_out, nomeEdificio, codiceOrchestra, via, nCivico, cap, citta) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

1. **Registrare l’acquisto di un nuovo biglietto per una data del concerto da parte di un cliente**

Se il cliente non è registrato lo registriamo:

INSERT INTO Cliente (cf, nome, cognome, numTel, email, età)

VALUES (?,?,?,?,?,?,?)

INSERT INTO Biglietto (codBigl, zona, costo, codOrch, giornoConc)

VALUES (?,?,?,?,?,?,?,?)

INSERT INTO Acquista (cf, dataAcq, oraAcq, metodoPagamento, codSconto, DescrSconto, percSconto, codBigl)

VALUES (?, CURRENT\_DATE, CURRENT\_TIME, ?, ?, ?, ?, ?);

1. **Cercare la città e il giorno dei concerti che si terranno in una certa città, esplicitando anche nome e cognome degli orchestrali**

select c.citta, c.giorno, c.titolo, o.nome, o.cognome, o.ruolo

from concerto c

join inclusione i on c.codiceOrchestra=i.codiceOrchestra

join orchestrali o on i.matricolaOrchestrali=o.matricola

where c.citta='verona'

1. **Mostrare tutte le serate in cui c’è sold out, specificando il numero dei biglietti venduti**   
   SELECT

c.giorno, c.ora, c.titolo,

COUNT(a.codice\_biglietto) AS Biglietti\_venduti

FROM

CONCERTO c

JOIN BIGLIETTO b ON c.codiceOrchestra = b.codiceOrchestra AND c.giorno = b.giornoConcerto

JOIN ACQUISTA a ON b.codice\_biglietto = a.codice\_biglietto

WHERE

c.sold\_out = 'sì'

GROUP BY

c.giorno,

c.ora,

c.titolo;

1. **Seleziona i musicisti presenti al concerto in una certa orchestra**

SELECT i.\*

FROM Inclusione i

JOIN Orchestrali o ON o.matricolaOrchestrali = i.matricolaOrchestrali

WHERE i.codiceOrchestra = ?;

1. **Mostrare l’elenco della scaletta dei brani di un concerto in cui è presente un certo direttore musicale**

SELECT

rb.ordine\_Brani, br.titolo, br.durata, co.titolo, co.giorno, co.citta, orr.nome, orr.cognome

FROM

rappresentati rb

JOIN CONCERTO co ON rb.codiceOrchestra = co.codiceOrchestra

AND rb.giornoConcerto = co.giorno

JOIN BRANO br ON rb.codiceBrano = br.codiceBrano

JOIN ORCHESTRA orc ON co.codiceOrchestra = orc.codiceOrchestra

JOIN inclusione inc ON orc.codiceOrchestra = inc.codiceOrchestra

JOIN ORCHESTRALI orr ON inc.matricolaOrchestrali = orr.matricola

WHERE

orr.nome = '?' AND

orr.cognome = '?’

ORDER BY

rb.ordine\_Brani;

1. **Mostrare a che concerto può andare una persona con un certo biglietto**   
   SELECT

c.giorno AS Giorno\_concerto,

c.ora AS ora\_concerto,

c.titolo AS titolo\_concerto,

c.descrizione AS descrizione\_concerto

FROM

BIGLIETTO b

JOIN CONCERTO c ON b.codiceOrchestra = c.codiceOrchestra AND b.giornoConcerto = c.giorno

WHERE

b.codice\_biglietto = ?;

1. **Aumentare il prezzo di tutti i biglietti di 5 euro**

UPDATE Biglietto   
SET costo=costo+'5'

**CAPITOLO 4 – PROGETTAZIONE DELL’APPLICAZIONE**

**DESCRIZIONE DELL'ARCHITETTURA DELL'APPLICAZIONE REALIZZATA**

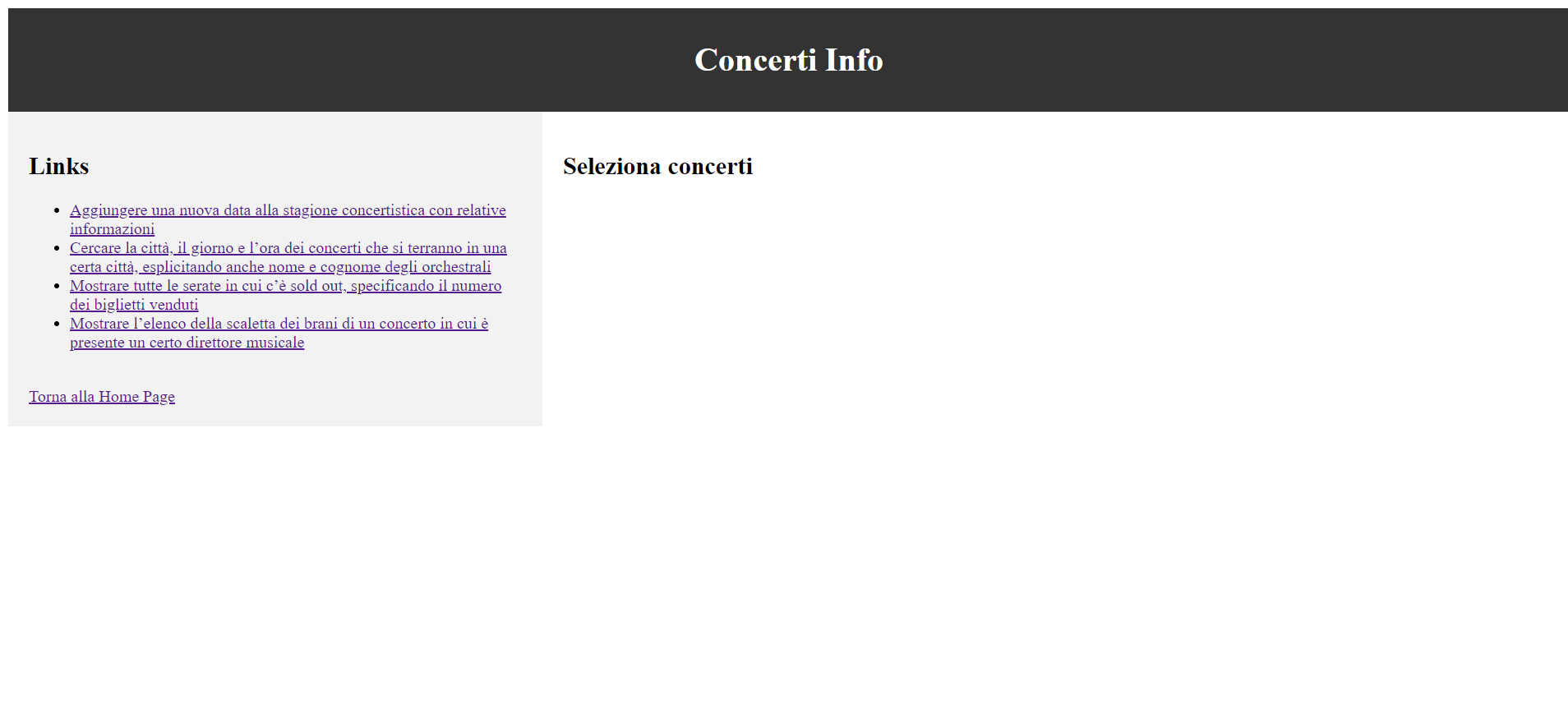
L’applicazione web per interfacciarsi al database è stata fatta con java script e php, aggiungendo parti in html e css per la grafica.

Essa si occupa di gestire le pagine dei concerti, dei biglietti e dei musicisti in cui all’interno si possono fare diverse operazioni.



Figura : schermata principale dell'applicazione

Nelle diverse pagine sono presenti dei link che portano a diverse operazioni che puoi fare (ovvero le query scritte nella relazione).



Questo infatti è un esempio di un’operazione che si può fare nella pagina concerto.



