

# PROGETTO BASI

## Basi di dati per la gestione di un museo

**Davide Lorenzon 2101075**

**Christian Libralato 2101047**

<b>1. ABSTRACT.....</b>	<b>1</b>
<b>2. ANALISI DEI REQUISITI.....</b>	<b>1</b>
<b>3. PROGETTAZIONE CONCETTUALE.....</b>	<b>3</b>
<b>4. PROGETTAZIONE LOGICA.....</b>	<b>4</b>
4.1 Analisi delle ridondanze (Visita_guidata).....	4
4.2 Analisi delle ridondanze (Guida).....	5
4.3 Eliminazione Generalizzazioni.....	8
4.4 Schema relazionale.....	9
<b>5. Implementazione in PostgreSQL e definizione delle query.....</b>	<b>10</b>
5.1 Definizione delle Query.....	10
• Query 1:.....	10
• Query 2:.....	11
• Query 3:.....	12
• Query 4:.....	12
• Query 5:.....	13
5.2 Definizione Indice.....	13
<b>6. Applicazione software.....</b>	<b>15</b>

## 1. ABSTRACT

Il progetto mira a creare un sistema di basi di dati per la gestione e monitoraggio di un museo. Con particolare attenzione a opere d'arte, relativi artisti, esposizioni e visite guidate, e figure dedicate, inoltre vi è un sistema di gestione dei biglietti venduti, questi si dividono in biglietti a ingresso libero e biglietti per una visita guidata.

Il sistema pone particolare attenzione sulla distinzione tra esposizioni passate e in corso, di quelle in corso si prevede di salvare anche l'informazione del curatore, che è una guida.

Il museo prevede la presenza di esposizioni intese come insieme di opere, le esposizioni hanno un argomento di riferimento e prevedono visite guidate tenute da un guida, ognuna delle quali è specializzata su un argomento in particolare.

Questa base di dati si prefigge di garantire un'organizzazione adatta alle necessità e a favorire un'analisi delle informazioni più efficiente

## 2. ANALISI DEI REQUISITI

In questa sezione vengono esposti i requisiti della base di dati.

**Artista:** L'artista creatore di un artefatto è identificato dal suo nome, cognome; contiene le seguenti informazioni:

- Nome
- Cognome
- Data di nascita
- Data di morte, se nota

**Artefatti:** Elemento di un'esposizione, viene creato da uno o più autori (se noti), a essi sono associate le seguenti informazioni :

- Codice univoco, che funge da identificatore
- Nome
- Tipologia (opere d'arte, reperti storici, ecc...)
- Esposizione in cui è esposto

**Esposizioni:** Consistono nelle mostre di diversi artefatti, si tengono in una specifica area e in determinato periodo definito da data di inizio e fine; Area e Data di inizio fungono da identificatore. Possiedono i seguenti dati:

- Nome
- Area
- Inizio
- Fine
- Argomento trattato dalla esposizione

Si mantiene lo storico delle esposizioni distinguendo tra un'esposizione in corso o una passata

**In Corso** Esposizioni attive al momento, devono essere gestite da un curatore che è una guida

- Curatore dell'esposizione, un curatore è una guida e non può occuparsi di più di un'esposizione per volta

**Passata** Esposizioni concluse

**Guide.** Le guide sono preparate su un argomento in particolare, sono identificate dal codice fiscale, presentano le seguenti informazioni:

- Codice Fiscale, che funge da identificatore
- Il numero di guide effettuate
- Argomento su cui sono specializzati

**Argomenti:** caratterizzano ciò che viene esposto in una certa esposizione e ciò che viene spiegato in una visita guidata

- Nome, che funge da identificatore

**Visite guidate:** rappresentano una visita guidata presso un'esposizione, sono caratterizzate da:

- ID, che funge da identificatore
- Data della visita
- Turno (mattina o pomeriggio) in cui si svolge
- Partecipanti alla guida
- Esposizione oggetto della visita guidata

Per ragioni logiche una guida non può presentare più visite guidate contemporaneamente (stessa data e turno)

**Biglietti.** biglietti venduti, si registrano i seguenti dati:

- ID univoco
- data di acquisto
- email fornita in fase di acquisto

Un biglietto può essere un biglietto ad accesso libero oppure un biglietto per visita guidata

**Ingresso libero.** in aggiunta alle informazioni del biglietto si ha:

- Data in cui è valido

**Ingresso guidato.** in aggiunta alle informazioni del biglietto vanno salvate le informazioni relative alla visita guidata di riferimento

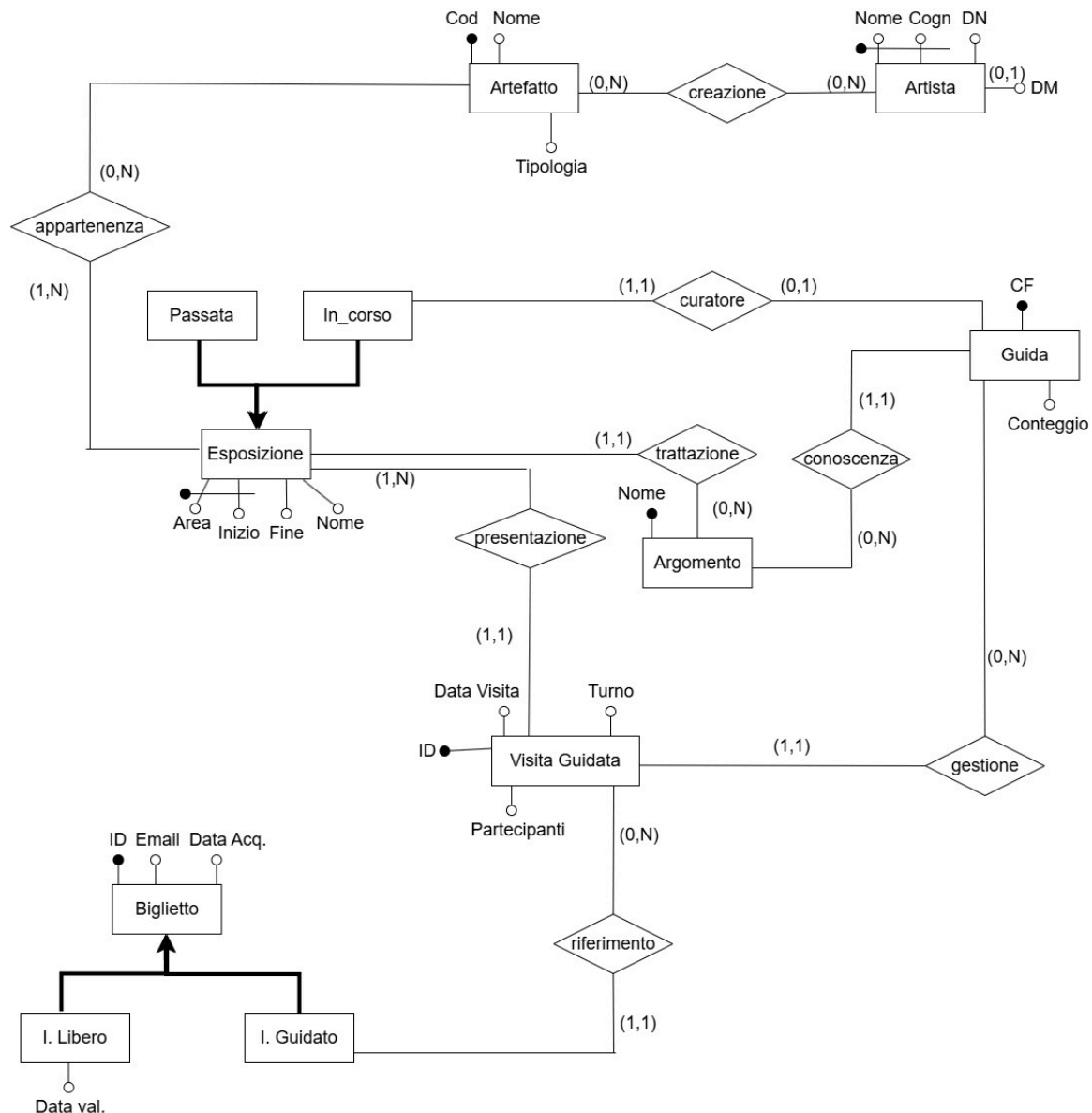
- Visita a cui il biglietto si riferisce

### 3. PROGETTAZIONE CONCETTUALE

Il fulcro del sistema di basi di dati sono le esposizioni e le visite guidate. Una visita guidata è tenuta da una e una sola guida in un determinato giorno, di mattina o di pomeriggio e in una certa esposizione. Ogni guida ha un argomento in cui ha conoscenza più vasta.

Per quanto concerne le esposizioni invece, esse rappresentano un insieme di artefatti che vengono esposti in un' area del museo, le esposizioni possono trattare di diversi argomenti e ospitare diverse visite guidate.

La gestione dei biglietti invece si basa su una suddivisione in biglietti a ingresso libero, che permettono di visitare tutte le esposizioni del museo, e biglietti a ingresso guidato che permettono di prendere parte alla visita guidata a cui si riferiscono.



Vincoli non rappresentabili nell' ER sono:

- Un'esposizione, una volta terminata, viene spostata dalle esposizioni in corso a quelle passate, curandosi di spostare le informazioni in esposizioni passate e rimuovendo l'informazione dell'esposizione corrente dagli artefatti passandola su quelle passate. Tale operazione sarebbe gestibile con un trigger, argomento non trattato a lezione e quindi non implementato
- La data di inizio di un'esposizione deve essere antecedente alla data di fine della stessa, gestita con un CHECK
- Data visita guidata sia coerente con le date dell'esposizione, non totalmente gestibile, si può usare un check per controllare data di visita successiva a data inizio (dato che fa parte della chiave esterna di visita) ma non si può controllare data fine senza un trigger. Quindi non controlliamo la coerenza con la data di fine.
- Una guida non può tenere più di una visita in una certa data e uno specifico turno, gestito con uno UNIQUE
- Una volta inserito un biglietto Ingresso guidato l'attributo partecipanti nella visita guidata relativa va incrementato di 1, richiederebbe l'uso di un trigger, non implementato per le medesime ragioni spiegate prima

## 4. PROGETTAZIONE LOGICA

### 4.1 Analisi delle ridondanze (Visita\_guidata)

L'attributo Numero\_Partecipanti in Visite\_guidate che memorizza il numero di partecipanti che hanno acquistato i biglietti per quella visita guidata è ridondante.

Può essere ricavato dalla relazione Riferimento che si collega con Ingresso\_Guidato

Questo attributo viene modificato ogni volta che viene acquistato un biglietto e si suppone che venga visualizzato ogni 30 minuti per monitorare la numerosità dei partecipanti.

Tutto ciò si riassume nelle seguenti 2 operazioni:

- Operazione 1)  
Memorizzare un biglietto per una visita guidata 200 volte al giorno
- Operazione 2)  
Visualizza i biglietti venduti, 48 volte al giorno

Assumiamo i volumi della base di dati come segue.

I risultati dell'analisi sono legati anche alla scelta arbitraria del numero di volte che eseguiamo le operazioni .

Concetto	Costrutto	Volume
Visita GUIDATA	E	10
Riferimento	R	500
Ingresso GUIDATO	E	500

La seguente analisi serve a valutare la convenienza dell'attributo ridondante:

**Con Ridondanza :**

- Operazione 1)

Concetto	Costrutto	Accessi	Tipo	
Visita GUIDATA	E	1	L	*200
Visita GUIDATA	E	1	S	*200
Riferimento	E	1	S	*200
Ingresso GUIDATO	R	1	S	*200

- Operazione 2)

Concetto	Costrutto	Accessi	Tipo	
Visita GUIDATA	E	1	L	*48

Considerando le operazioni di scrittura con un costo doppio rispetto a quelle di lettura si ha:  
Costo Totale:  $200 \times 6 + 200 + 48 = 1448$

**Senza ridondanza :**

- Operazione 1

Riferimento	E	1	S	*200
Ingresso GUIDATO	R	1	S	*200

- Operazione 2

Concetto	Costrutto	Accessi	Tipo	
Visita GUIDATA	E	1	L	*48
Riferimento	R	50	L	*48

Considerando le operazioni di scrittura con un costo doppio rispetto a quelle di lettura si ha:  
Costo Totale:  $200 \times 4 + 48 \times 51 = 3248$

Dall'analisi si evince come sia vantaggioso in termini di costo mantenere la ridondanza.

## 4.2 Analisi delle ridondanze (Guida)

L'attributo Conteggio, che indica il numero di visite guidate effettuate da una certa guida è ridondante in quanto si può ricavare dalla relazione Gestione che si collega con Visita GUIDATA

Questo attributo viene modificato ogni volta che una guida viene programmata una visita guidata.

Tutto ciò si riassume nelle seguenti 2 operazioni:

- Operazione 1)  
Memorizzare una visita guidata, effettuato 2 volte al giorno
- Operazione 2)  
Conteggio guide assegnate, controllato una volta al giorno

Assumiamo i volumi della base di dati come segue.

I risultati dell'analisi sono legati anche alla scelta arbitraria del numero di volte che eseguiamo le operazioni .

Concetto	Costrutto	Volume
Visita GUIDATA	E	10
Gestione	R	10
Guida	E	5

La seguente analisi serve a valutare la convenienza dell'attributo ridondante:

**Con Ridondanza :**

- Operazione 1)

Concetto	Costrutto	Accessi	Tipo	
Visita GUIDATA	E	1	S	*2
Gestione	E	1	S	*2
Guida	E	1	S	*2
Guida	R	1	L	*2

- Operazione 2)

Concetto	Costrutto	Accessi	Tipo	
Guida	E	1	L	*1

Considerando le operazioni di scrittura con un costo doppio rispetto a quelle di lettura si ha:

Costo Totale:  $2 \times 6 + 2 + 1 = 15$

**Senza ridondanza :**

- Operazione 1

Concetto	Costrutto	Accessi	Tipo	
Visita GUIDATA	E	1	S	*2
Gestione	R	1	S	*2

- Operazione 2

Concetto	Costrutto	Accessi	Tipo	
Guida	E	1	L	*1
Gestione	R	2	L	*1

Considerando le operazioni di scrittura con un costo doppio rispetto a quelle di lettura si ha:

Costo Totale:  $2 \times 4 + 1 + 1 \times 2 = 11$

Da questa analisi di costo risulta vantaggioso in termini di costo togliere la ridondanza.

## Dizionario dei dati

Entità			
Entità	Descrizione	Attributi	Identificatore
Artista	Autori di opere d'arte, reperti o affini	Nome, Cognome, Data_Nascita, Data_Morte	Nome, Cognome
Artefatto	Oggetto di interesse che può essere esposto	Codice, Nome, Tipologia	Codice
Esposizione	Esposizione	Nome, Area, Inizio, Fine	Area, Inizio
Passata	Esposizione conclusa		
In_Corso	Esposizione corrente		
Argomento	Argomento di interesse storico	Nome	Nome
Guida	Guida per le visite guidate	CF	CF
Visita_Guidata	Visita guidata di un'esposizione	ID, Data_Visita, Turno Numero_Partecipanti	ID
Biglietto	Biglietto per entrare nel museo	ID, data_Acquisto, mail	ID
Ingresso_Libero	Biglietto ad ingresso libero	Data_Visita	
Ingresso_Guidato	Biglietto per una visita guidata		

Relazioni			
Relazione	Descrizione	Componente	Attributi
Creazione	Creazione di un artefatto da parte di autori	Artefatto, Artista	
Appartenenza	Appartenenza di un artefatto a un'esposizione	Artefatto, Esposizione	
Trattazione	Argomenti dell'esposizione	Argomento, Esposizione	

Relazioni			
Conoscenza	Argomento in cui una guida è specializzata	Argomento, Guida	
Curatore	Curatore di una mostra in corso	Guida, In_corso	
Presentazione	Visite guidate in un'esposizione	Esposizione, Visita_Guidata	
Gestione	Visite guidate effettuate da una guida	Guida, Visita_Guidata	
Riferimento	Riferimento di un biglietto a una visita guidata	Visita_Guidata, Ingresso_Guidato	

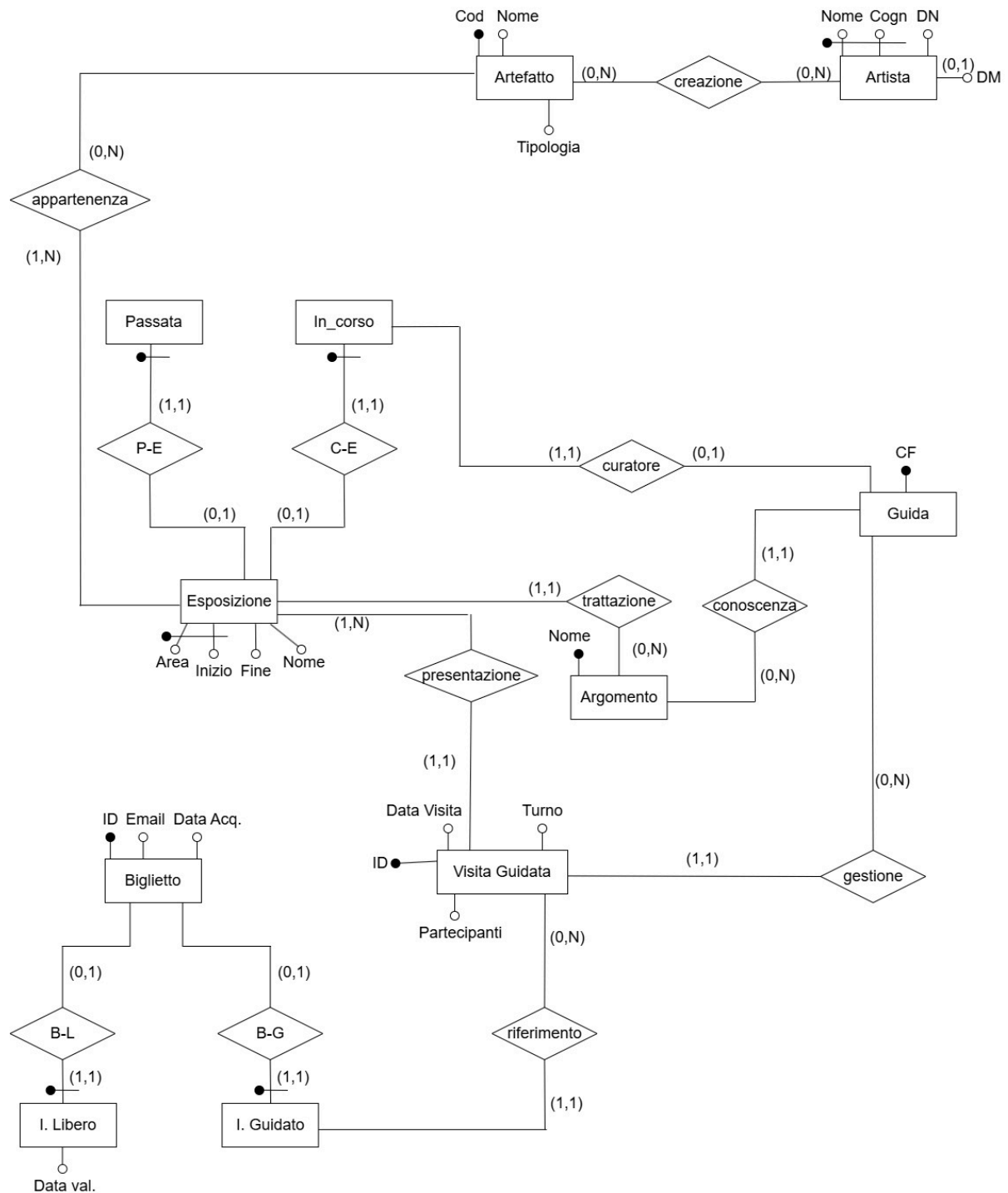
### 4.3 Eliminazione Generalizzazioni

Le generalizzazioni introdotte vengono rimosse nel processo di ristrutturazione dello schema concettuale al fine di rendere lo schema implementabile come modello relazionale.

**Biglietto:** La generalizzazione totale di Biglietto è stata rimossa inserendo le relazioni B-L e B-G rispettivamente per Ingresso\_Libero e Ingresso\_Guidato, ciò ha permesso di mantenere la relazione funzionale Riferimento tra Ingresso\_Guidato e Visita\_Guidata e di ridurre la possibilità di avere campi nulli. Se si mantenesse una singola entità Biglietto, si avrebbero sempre dei campi nulli per differenziare le due tipologie di ingresso e salvare attributi specifici.

**Esposizione:** La generalizzazione totale di esposizione permette di mantenere uno storico, data la presenza una relazione in ingresso all'entità In\_Corso la generalizzazione è stata sostituita con l'inserimento di due relazioni P-E e C-E rispettivamente per Passata e In Corso.





## 4.4 Schema relazionale

Lo schema ristrutturato contiene solamente costrutti mappabili in corrispettivi dello schema relazionale - detto anche schema logico. Lo schema logico è rappresentato a seguire, dove l'asterisco dopo il nome degli attributi indica quelli che ammettono valori nulli.

- Artista(Nome, Cognome, DN, DM\*)
- Argomento(Nome)

- Guida(CF,Argomento)  
Guida.Argomento->Argomento.Nome
- Artefatto(Codice, Nome, Tipologia)
- Esposizione(Nome,Area, Inizio, Fine , Argomento)  
-Esposizione. Argomento->Argomento.Nome
- In\_Corso(Area,Inizio,Curatore)  
-In\_Corso.(Area,Inizio,Curatore) -> Esposizione.(Area,Inizio), Guida.CF
- Passata(Area,Inizio)  
-Passata.(Area,Inizio) -> Esposizione.(Area,Inizio)
- Appartenenza(Area,Inizio,Artefatto)  
Appartenenza(Area,Inizio,Artefatto)->Esposizione.(Area,Inizio),Artefatto.Codice
- Visita\_Guidata( ID, Area, Inizio, Guida, Data\_Visita, Turno, Partecipanti)  
Visita\_Guidata.(Guida,Area,Inizio)->Guida.CF, Esposizione(Area,Inizio)
- Biglietto(ID,Data\_Acq.,EMail)
- Ingresso\_Guidato(ID,IDVG)  
Ingresso\_Guidato.(ID, IDVG)->Biglietto.ID, Visita\_Guidata.ID
- Ingresso\_Libero(ID, Data\_validità)  
Ingresso\_Libero.ID->Biglietto.ID
- Creazione(Codice, Nome, Cognome)  
Creazione.(Codice, Nome, Cognome)->Artefatto.Codice,Artista.(Nome, Cognome)

## 5. Implementazione in PostgreSQL e definizione delle query

Il file create.sql permette la creazione e il popolamento delle tabelle. In esso sono inoltre contenute cinque query, una vista e la definizione di un indice per velocizzare l'esecuzione di una di esse.

### 5.1 Definizione delle Query

- **Query 1:**

Trovare le esposizioni in corso che hanno almeno un certo numero di artefatti esposti appartenenti ad un determinato autore e visualizzare il numero di artefatti.

Parametri:

min = numero artefatti,

nom = nome autore,

cogn = cognome autore

Valori:

min= 1  
nom = 'Leonardo'  
cogn = 'da Vinci'  
si ottiene:

-----  
SELECT A.Area, A.Inizio, COUNT(\*) AS num\_esposizioni  
FROM Creazione C, (Artefatto JOIN Appartenenza ON Artefatto.codice=Appartenenza.Artefatto) A  
WHERE C.Codice = A.Codice AND C.Nome = 'Leonardo' AND C.Cognome = 'da Vinci'  
GROUP BY A.Area, A.Inizio  
HAVING COUNT(\*) >= 1;  
-----

	area character varying (32)	inizio date	num_esposizioni bigint
1	Area1	2023-01-15	2
Total rows: 1    Query complete 00:00:00.242			

- **Query 2:**

Contare quanti biglietti ad ingresso guidato per un esposizione di un certo argomento sono stati venduti indicando anche la data di acquisto.

Parametri:  
arg = argomento  
Valori:  
arg = 'Rinascimento'  
si ottiene:

-----  
SELECT IG.Data\_Acq, COUNT(\*) AS Num\_biglietti  
FROM (Ingresso\_Guidato JOIN Biglietto ON Ingresso\_Guidato.id=Biglietto.id) IG, Visita\_Guidata VG,  
Esposizione E  
WHERE VG.Area = e.Area AND VG.Inizio = E.Inizio AND IG.IDVG = VG.ID  
AND E.Argomento ='Rinascimento'  
GROUP BY IG.Data\_Acq;  
-----

	data_acq date	num_biglietti bigint
1	2023-02-08	2
2	2023-10-06	8
Total rows: 2    Query complete 00:00:00.264		

- **Query 3**

Contare il numero di visite guidate che hanno tenuto visite guidate in esposizioni di un certo argomento a partire da una certa data e calcolare la media dei loro partecipanti

Parametri:

arg = argumento,

```
dt = data
```

Valori:

```
dt = '2015-12-01'
```

```
arg = 'Rinascimento'
```

si ottiene:

```
SELECT AVG(A.Partecipanti) AS partecipanti_medi, COUNT(*) AS num_visite
FROM (Visita_guidata VG JOIN Esposizione E ON VG.Area = E.Area AND VG.Inizio = E.inizio) A
WHERE A.Data_visita >= '2015-12-01' AND A.Argomento = 'Rinascimento';
```

	partecipanti_medici numeric	num_visite bigint
1	1.2500000000000000	8

Total rows: 1      Query complete 00:00:00.251

- **Query 4:**

Trovare il Codice Fiscale delle guide che hanno tenuto più visite guidate su un certo argomento, indicare anche il numero di visite.

Parametri:

arg = argomento

Valori:

arg = 'Rinascimento'

si ottiene:

```
CREATE VIEW Conteggio_Visite AS (
SELECT Guida, Argomento, COUNT(*) AS NumVisite
FROM Esposizione E JOIN Visita_Guidata V ON (E.area=V.Area AND E.Inizio=V.Inizio)
GROUP BY Guida, Argomento
);
```

[illegible]

	cf character (32)	numvisite bigint
1	RSSMRA85M10H50...	8

Total rows: 1    Query complete 00:00:00.355

### ● Query 5:

Calcolare la media del numero di artefatti di un determinato artista presenti in ogni area.

Parametri:

nom = nome,

cogn = cognome

Valori:

nom = 'Leonardo'

cogn = 'da Vinci'

si ottiene:

-----

```
SELECT AVG(Num) AS media_artefatti
FROM (SELECT COUNT(*) AS Num
FROM Creazione C, (Artefatto Ar JOIN Appartenenza Ap ON Ar.Codice=Ap.Artefatto) A
WHERE A.Codice = C.Codice AND C.Nome = 'Leonardo' AND C.Cognome = 'da Vinci'
GROUP BY A.Area);
```

-----

	media_artefatti numeric
1	2.0000000000000000

Total rows: 1    Query complete 00:00:00.240

## 5.2 Definizione Indice

Abbiamo scelto di implementare il seguente indice:

```
CREATE INDEX index_visits ON Visita_Guidata ( Data_Visita );
```

Al fine di migliorare le prestazioni della query 3, dato che la struttura ad albero, struttura di default, permette di migliorare l'operazione di confronto (>) tra le date delle visite guidate.

Prima dell'indice: 66 millisecondi

Query Query History Scratch Pad x

```
1 SELECT AVG(A.Partecipanti), COUNT(*)
2 FROM (Visita_guidata VG JOIN Esposizione E ON VG.Area = E.Area AND VG.Inizio = E.inizio)
3 WHERE A.Data_visita >= '2020-01-10' AND A.Argomento = 'Rinascimento'
4
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	avg numeric	count bigint
1	1.2500000000000000	8

Successfully run. Total query runtime: 66 msec. 1 rows affected. X

Total rows: 1 Query complete 00:00:00.066 LF Ln 4, Col 1

Dopo l'indice: 51 millisecondi

primo/postgres@prova

Query Query History Scratch Pad x

```
1 SELECT AVG(A.Partecipanti), COUNT(*)
2 FROM (Visita_guidata VG JOIN Esposizione E ON VG.Area = E.Area AND VG.Inizio = E.inizio)
3 WHERE A.Data_visita >= '2020-01-10' AND A.Argomento = 'Rinascimento'
4
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	avg numeric	count bigint
1	1.2500000000000000	8

Successfully run. Total query runtime: 51 msec. 1 rows affected. X

Total rows: 1 Query complete 00:00:00.051 LF Ln 1, Col 1

## **6. Applicazione software**

Il file QueryMenu.c contiene il codice in C che offre un'interfaccia da linea di comando per connettersi al database ed eseguire le query precedentemente elencate, scegliendo dall'apposito menù.

Inoltre è possibile scegliere i parametri delle query indicando i valori da riga di comando o scegliendo in base da valori salvati nel database.