

Baza de date pentru gestionarea unui muzeu de artă

Țopai Ioan-Alexandru 341A3

Cuprins

| | |
|--|---|
| Descrierea Temei | 3 |
| Descrierea bazei de date: | 3 |
| Diagrama bazei de date: | 3 |
| Structura tabelelor | 3 |
| Descrierea constrângerilor de integritate | 5 |
| Descrierea procedurilor și funcțiilor | 5 |
| Descrierea aplicației: | 6 |
| Prezentarea modului în care se face conexiunea cu baza de date | 6 |
| Capturi de ecran pentru interfețe și rapoarte | 7 |
| Concluzii | 8 |
| Bibliografie | 8 |

Descrierea Temei

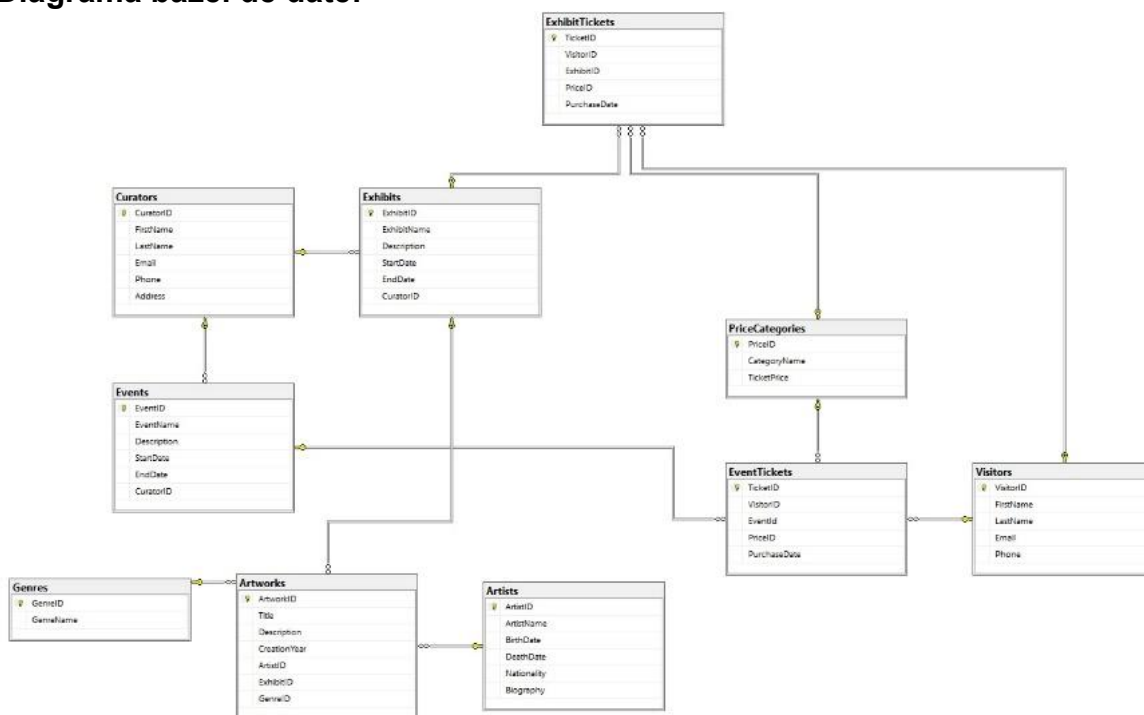
Proiectul constă în dezvoltarea unei baze de date pentru gestionarea unui muzeu de artă, cu scopul de a organiza și gestiona informații legate de operele de artă, artiști, angajați, vizitatori, precum și gestionarea vânzării de bilete pentru evenimente și expoziții.

Scopul proiectului:

- **Administrarea Operei de Artă:** Urmărirea și organizarea informațiilor despre operele de artă expuse în muzeu, inclusiv detalii despre artist, anul de creație, genul artistic și detaliile expoziției curente.
- **Gestionarea Artiștilor:** Colectarea și gestionarea datelor despre artiști, inclusiv biografii, date de naștere și deces, naționalitate, și operele lor de artă.
- **Evidența Angajaților:** Monitorizarea informațiilor despre curatori și alți angajați, inclusiv date de contact și informații de gestionare.
- **Vânzarea de Bilete:** Gestionarea vânzării de bilete pentru evenimente și expoziții, inclusiv categorii de prețuri și detaliile achiziționării.
- **Interacțiune cu Vizitatorii:** Urmărirea informațiilor despre vizitatori, inclusiv date de contact.

Descrierea bazei de date:

Diagrama bazei de date:



Structura tabelor

1. Genres
 - GenreID (INT, Primary Key)
 - GenreName (VARCHAR(100))
2. Curators
 - CuratorID (INT, Primary Key)

- FirstName (VARCHAR(255)) NOT NULL
 - LastName (VARCHAR(255)) NOT NULL
 - Email (VARCHAR(255))
 - Phone (VARCHAR(20)) NOT NULL
 - Address (VARCHAR(255))
3. Exhibits
- ExhibitID (INT, Primary Key)
 - ExhibitName (VARCHAR(255)) NOT NULL
 - Description (TEXT)
 - StartDate (DATE)
 - EndDate (DATE)
 - CuratorID (INT, Foreign Key to Curators.CuratorID) NOT NULL
4. Artists
- ArtistID (INT, Primary Key)
 - ArtistName (VARCHAR(255)) NOT NULL
 - BirthDate (DATE)
 - DeathDate (DATE)
 - Nationality (VARCHAR(100))
 - Biography (VARCHAR(MAX))
5. Artworks
- ArtworkID (INT, Primary Key)
 - Title (VARCHAR(255)) NOT NULL
 - Description (TEXT)
 - CreationYear (INT)
 - ArtistID (INT, Foreign Key to Artists.ArtistID) NOT NULL
 - ExhibitID (INT, Foreign Key to Exhibits.ExhibitID) NOT NULL
 - GenreID (INT, Foreign Key to Genres.GenreID) NOT NULL
6. Visitors
- VisitorID (INT, Primary Key)
 - FirstName (VARCHAR(100)) NOT NULL
 - LastName (VARCHAR(100)) NOT NULL
 - Email (VARCHAR(255))
 - Phone (VARCHAR(20))
7. PriceCategories
- PriceID (INT, Primary Key)
 - CategoryName (VARCHAR(100)) NOT NULL
 - TicketPrice (DECIMAL(10, 2)) NOT NULL
8. Events
- EventID (INT, Primary Key)
 - EventName (VARCHAR(255)) NOT NULL
 - Description (TEXT)
 - StartDate (DATETIME)
 - EndDate (DATETIME)
 - CuratorID (INT, Foreign Key to Curators.CuratorID) NOT NULL
9. EventTickets
- TicketID (INT, Primary Key)
 - VisitorID (INT, Foreign Key to Visitors.VisitorID)
 - EventID (INT, Foreign Key to Events.EventID) NOT NULL
 - PriceID (INT, Foreign Key to PriceCategories.PriceID) NOT NULL

- PurchaseDate (DATETIME) NOT NULL
10. ExhibitTickets
- TicketID (INT, Primary Key)
 - VisitorID (INT, Foreign Key to Visitors.VisitorID)
 - ExhibitID (INT, Foreign Key to Exhibits.ExhibitID) NOT NULL
 - PriceID (INT, Foreign Key to PriceCategories.PriceID) NOT NULL
 - PurchaseDate (DATETIME) NOT NULL

Descrierea constrângerilor de integritate

Constrângeri Cheie Primară (Primary Key Constraints):

Fiecare tabel are o cheie primară unică care identifică în mod unic fiecare înregistrare. De exemplu, ArtworkID în tabela Artworks sau VisitorID în tabela Visitors.

Constrângeri Cheie Secundară (Foreign Key Constraints):

Relațiile dintre tabele sunt definite prin chei străine. De exemplu, ArtistID în tabela Artworks este o cheie străină care se referă la ArtistID din tabela Artists.

Constrângeri NOT NULL (Not Null Constraints):

Aceste constrângeri asigură că anumite câmpuri nu pot avea valori nule. De exemplu, FirstName și LastName în tabela Visitors trebuie să nu fie nule.

Descrierea procedurilor și funcțiilor

1. Procedura: GetArtworkCountsPerGenre

Scop:

Obținerea numărului de opere de artă pentru fiecare artist și gen artistic, cu opțiunea de a filtra rezultatele în funcție de un număr minim de opere.

Parametri:

@numberOfArts (INT): Numărul minim de opere necesar pentru a fi inclus în rezultate.

Descriere:

- Se realizează o interogare care se alătură tablei Artworks cu tablele Artists și Genres.
- Se grupează rezultatele în funcție de artist și gen artistic.
- Se filtrează rezultatele pentru a include doar acele înregistrări cu un număr de opere mai mare decât @numberOfArts.
- Rezultatele sunt ordonate după artist și gen artistic.
-

2. Procedura: GetTotalSpentOnEventsInRange

Scop:

Obținerea sumei totale cheltuite de către vizitatori pentru evenimente într-un interval de ani specificat.

Parametri:

@p_StartYear (INT): Anul de început al intervalului.

@p_EndYear (INT): Anul de sfârșit al intervalului.

Descriere:

- Se realizează o interogare care se alătură tabelelor Visitors, EventTickets, PriceCategories, și Events.
- Se filtrează rezultatele pentru a include doar evenimentele achiziționate în intervalul de ani specificat.
- Rezultatele sunt grupate în funcție de vizitator și anul achiziției, calculând suma totală cheltuită pe bilete.
- Se exclud înregistrările cu valoarea sumei egale cu 0.
- Rezultatele sunt ordonate descrescător după anul achiziției și suma totală cheltuită.

3. Procedura: GetCuratorPerformanceByYear

Scop:

Obținerea performanței curatorilor pentru un an specificat, în funcție de numărul total de vizitatori și venitul total generat din vânzarea de bilete la expoziții și evenimente.

Parametri:

@p_Year (INT): Anul pentru care se dorește evaluarea performanței curatorilor.

Descriere:

- Se realizează o interogare care se alătură tabelelor Curators, Exhibits, ExhibitTickets, Events, și EventTickets.
- Se filtrează rezultatele pentru a include doar înregistrările corespunzătoare anului specificat.
- Se realizează o unificare (UNION ALL) a vizitatorilor care au achiziționat bilete pentru expoziții și evenimente.
- Rezultatele sunt grupate în funcție de curator, numărul total de vizitatori, și venitul total generat.
- Performanța este evaluată în funcție de venitul total, iar rezultatele sunt ordonate descrescător după venitul total.

Descrierea aplicației:

Prezentarea modului în care se face conexiunea cu baza de date

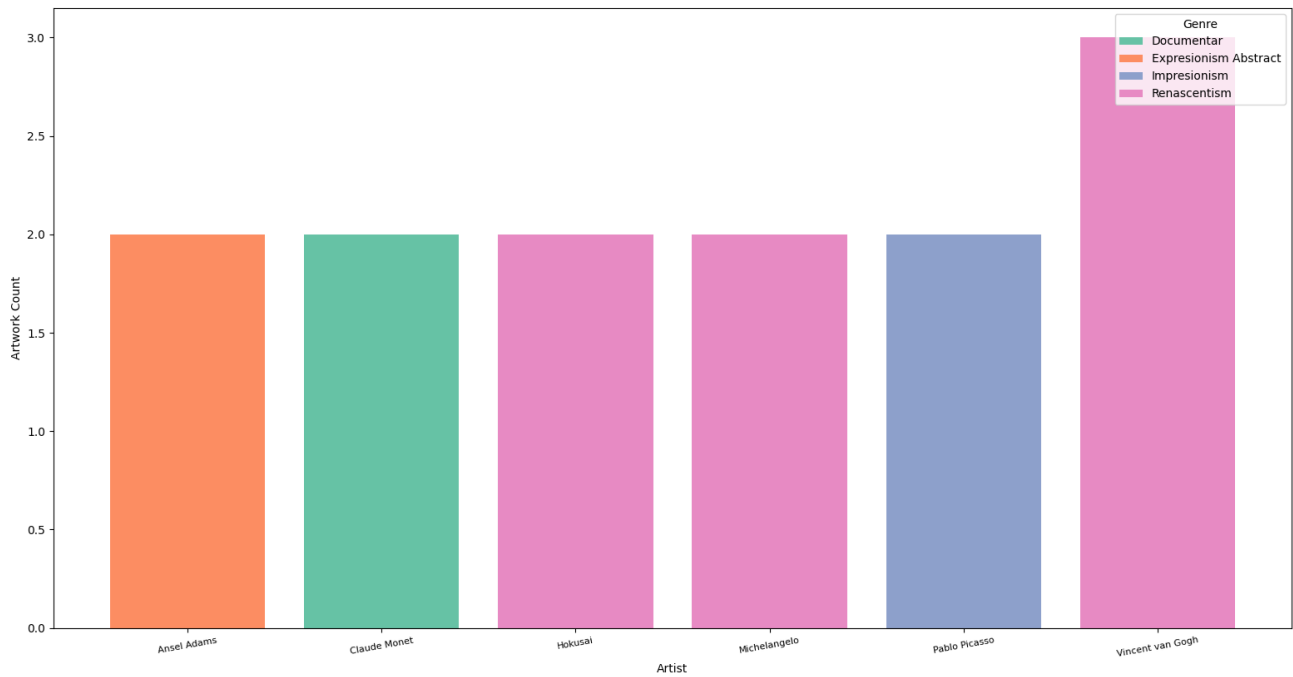
Constructorul clasei primește parametri necesari pentru a stabili conexiunea: **host** (numele serverului), **port** (numărul portului), și **database** (numele bazei de date).

openConnection(self):

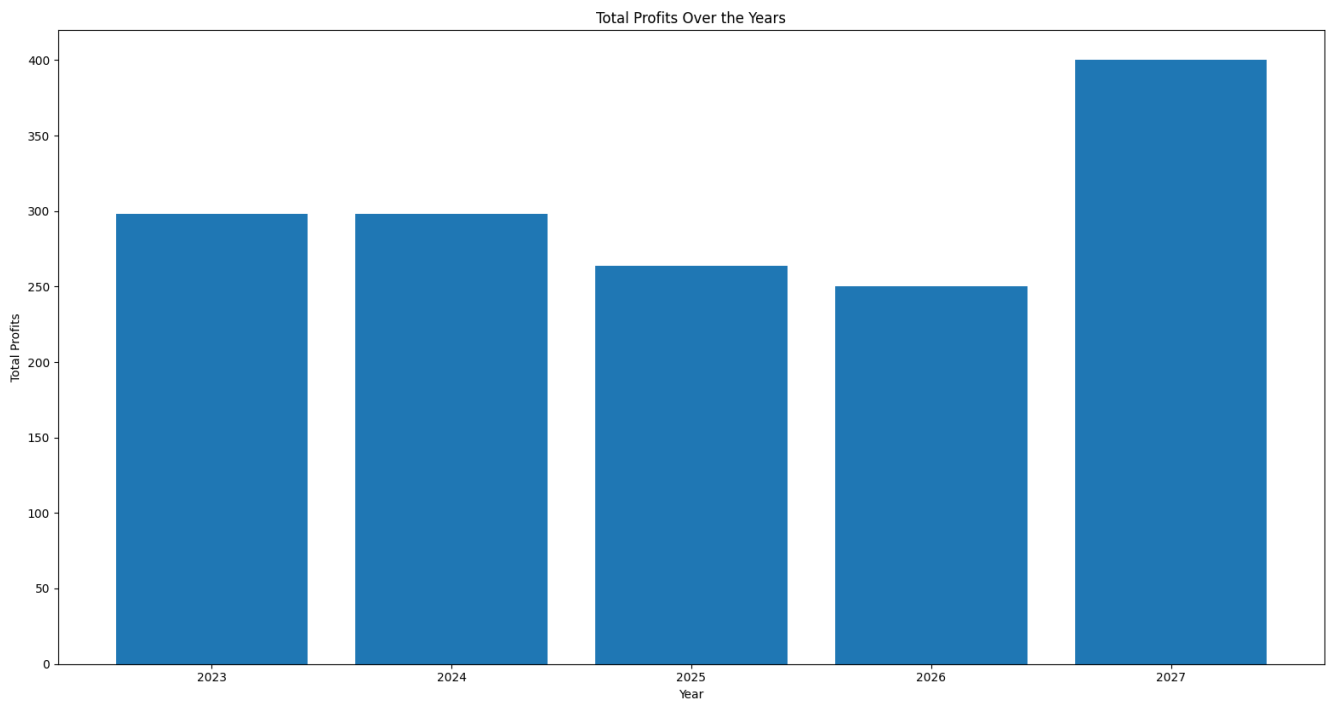
- Metoda deschide conexiunea către baza de date utilizând pyodbc.
- Se utilizează driverul ODBC 17 pentru SQL Server.
- Parametrii conexiunii sunt furnizați din atributele obiectului (host, port, database).
- Conexiunea este stabilită și un cursor este creat pentru execuția de interogări SQL.

Capturi de ecran pentru interfețe și rapoarte

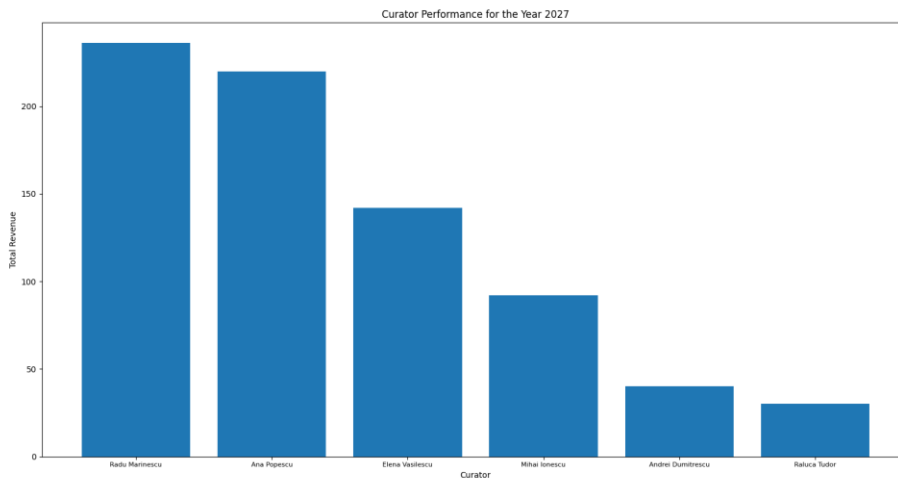
1. Procedura: GetArtworkCountsPerGenre



2. Procedura: GetTotalSpentOnEventsInRange



3. Procedura: GetCuratorPerformanceByYear



Concluzii

În concluzie, proiectul bazei de date pentru gestionarea unui muzeu de artă oferă o structură solidă pentru stocarea și gestionarea informațiilor legate de operele de artă, artiști și evenimente. Procedurile stocate furnizează funcționalități utile pentru evaluarea performanței și interacțiunea cu vizitatorii. Clasa de conexiune facilitează interacțiunea cu baza de date în mediul Python. În ansamblu, sistemul oferă o soluție eficientă și modulară pentru gestionarea unui muzeu de artă.

Bibliografie

1. <https://learn.microsoft.com/ro-ro/sql/connect/python/pyodbc/python-sql-driver-pyodbc?view=sql-server-2016>
2. <https://docs.python.org/3/>