Проектування БД за індивідуальними завданнями

Виконати системний аналіз вибраної вами предметної області:

1) Зробити опис предметної області – області автоматизації:

a) користувачі та їх вимоги до створюваної автоматизованої системи

b) документи, що циркулюють в предметній області,

c) правила за якими вони формуються

d) описати обмеження на інформацію, що повинна зберігатись в БД

2) Сформувати словник БД .

3) Визначити сутності предметної області, їх атрибути .

4) Визначити зв'язки між сутностями предметної області. Побудувати інфологічну модель.

5) Побудувати таблиці опису та зв’язки між ними. Зробити запити.

**Інформаційна система театру**

**1. Опис предметної області**

**a) Користувачі та їх вимоги до створюваної автоматизованої системи:**

**Адміністратори:**

* **Керування інформацією про вистави (створення, оновлення, видалення).**
* **Керування інформацією про акторів та персонал.**
* **Генерація звітів про продажі квитків та відвідуваність.**
* **Керування користувачами системи.**

**Касири:**

* **Продаж квитків.**
* **Внесення інформації про покупців.**
* **Перевірка наявності квитків.**

**Клієнти (глядачі):**

* **Перегляд репертуару.**
* **Бронювання та покупка квитків.**
* **Отримання інформації про вистави та акторів.**

**b) Документи, що циркулюють в предметній області:**

**Розклад вистав:**

* **Інформація про дати та часи проведення вистав.**

**Списки акторів:**

* **Інформація про акторів, які беруть участь у виставі.**

**Продані квитки:**

* **Інформація про продані квитки (номер квитка, місце, дата продажу).**

**Замовлення на квитки:**

* **Інформація про бронювання квитків (ім'я клієнта, контактні дані, кількість квитків).**

**Фінансові звіти:**

* **Інформація про доходи від продажу квитків.**

**c) Правила за якими формуються документи:**

* **Розклад вистав формується адміністратором театру на основі запланованих заходів.**
* **Списки акторів формуються на основі репертуару та розподілу ролей.**
* **Продані квитки фіксуються в системі при продажу квитка касиром.**
* **Замовлення на квитки фіксуються при бронюванні квитка клієнтом через систему.**
* **Фінансові звіти формуються автоматично на основі даних про продажі квитків.**

**d) Опис обмежень на інформацію, що повинна зберігатись в БД:**

* **Унікальність номерів квитків.**
* **Унікальність ID вистав та акторів.**
* **Обмеження на дати і часи проведення вистав (неперетинання).**
* **Валідація контактних даних клієнтів (правильність номерів телефонів, електронних адрес).**

**2. Словник БД**

|  |  |
| --- | --- |
| **Таблиця** | **Опис** |
| **Users** | **Користувачі системи(адміністратори, касири)** |
| **Performances** | **Вистави та розклад** |
| **Actors** | **Актори театру** |
| **Tickets** | **Продані квитки** |
| **Reservations** | **Замовлення на квитки** |
| **FinancialReports** | **Фінансові звіти** |

**3. Визначення сутностей та їх атрибутів**

**Users:**

**UserID (INT, PK)**

**Username (VARCHAR)**

**Password (VARCHAR)**

**Role (ENUM: 'admin', 'cashier')**

**Performances:**

**PerformanceID (INT, PK)**

**Title (VARCHAR)**

**Date (DATE)**

**Time (TIME)**

**Duration (INT)**

**Description (TEXT)**

**Actors:**

**ActorID (INT, PK)**

**Name (VARCHAR)**

**Bio (TEXT)**

**Birthdate (DATE)**

**Tickets:**

**TicketID (INT, PK)**

**PerformanceID (INT, FK)**

**SeatNumber (VARCHAR)**

**Price (DECIMAL)**

**SaleDate (DATETIME)**

**CustomerName (VARCHAR)**

**CustomerContact (VARCHAR)**

**Reservations:**

**ReservationID (INT, PK)**

**PerformanceID (INT, FK)**

**SeatNumber (VARCHAR)**

**CustomerName (VARCHAR)**

**CustomerContact (VARCHAR)**

**ReservationDate (DATETIME)**

**Status (ENUM: 'reserved', 'paid', 'cancelled')**

**FinancialReports:**

**ReportID (INT, PK)**

**Date (DATE)**

**TotalIncome (DECIMAL)**

**TotalTicketsSold (INT)**

**4. Визначення зв'язків між сутностями та побудова інфологічної моделі**

**Users (1) <----> (N) FinancialReports**

**Один користувач може створювати багато фінансових звітів.**

**FinancialReports.UserID посилається на Users.UserID.**

**Performances (1) <----> (N) Tickets**

**Одна вистава може мати багато квитків.**

**Tickets.PerformanceID посилається на Performances.PerformanceID.**

**Performances (1) <----> (N) Reservations**

**Одна вистава може мати багато замовлень.**

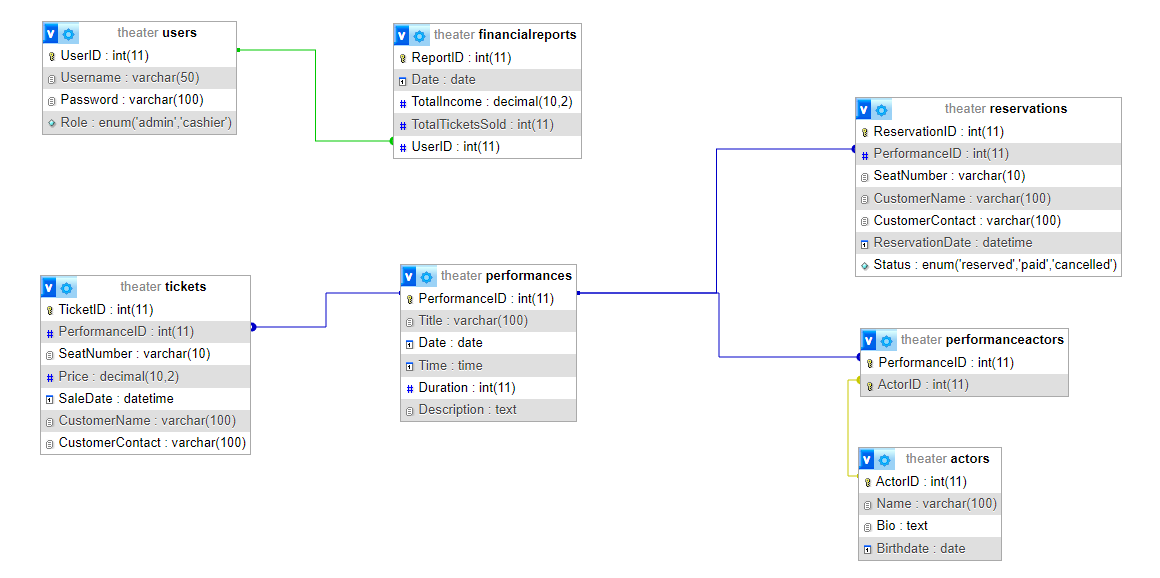
**Reservations.PerformanceID посилається на Performances.PerformanceID.**

**Performances (N) <----> (N) Actors (через PerformanceActors)**

**Одна вистава може мати багато акторів, і один актор може брати участь у багатьох виставах.**

**PerformanceActors.PerformanceID посилається на Performances.PerformanceID.**

**PerformanceActors.ActorID посилається на Actors.ActorID.**

****

**5. Побудова таблиць опису та зв’язки між ними**

**CREATE TABLE Users (**

**UserID INT AUTO\_INCREMENT PRIMARY KEY,**

**Username VARCHAR(50) NOT NULL,**

**Password VARCHAR(100) NOT NULL,**

**Role ENUM('admin', 'cashier') NOT NULL**

**);**

**CREATE TABLE Performances (**

**PerformanceID INT AUTO\_INCREMENT PRIMARY KEY,**

**Title VARCHAR(100) NOT NULL,**

**Date DATE NOT NULL,**

**Time TIME NOT NULL,**

**Duration INT NOT NULL,**

**Description TEXT**

**);**

**CREATE TABLE Actors (**

**ActorID INT AUTO\_INCREMENT PRIMARY KEY,**

**Name VARCHAR(100) NOT NULL,**

**Bio TEXT,**

**Birthdate DATE**

**);**

**CREATE TABLE PerformanceActors (**

**PerformanceID INT,**

**ActorID INT,**

**PRIMARY KEY (PerformanceID, ActorID),**

**FOREIGN KEY (PerformanceID) REFERENCES Performances(PerformanceID),**

**FOREIGN KEY (ActorID) REFERENCES Actors(ActorID)**

**);**

**CREATE TABLE Tickets (**

**TicketID INT AUTO\_INCREMENT PRIMARY KEY,**

**PerformanceID INT,**

**SeatNumber VARCHAR(10),**

**Price DECIMAL(10, 2),**

**SaleDate DATETIME,**

**CustomerName VARCHAR(100),**

**CustomerContact VARCHAR(100),**

**FOREIGN KEY (PerformanceID) REFERENCES Performances(PerformanceID)**

**);**

**CREATE TABLE Reservations (**

**ReservationID INT AUTO\_INCREMENT PRIMARY KEY,**

**PerformanceID INT,**

**SeatNumber VARCHAR(10),**

**CustomerName VARCHAR(100),**

**CustomerContact VARCHAR(100),**

**ReservationDate DATETIME,**

**Status ENUM('reserved', 'paid', 'cancelled') DEFAULT 'reserved',**

**FOREIGN KEY (PerformanceID) REFERENCES Performances(PerformanceID)**

**);**

**CREATE TABLE FinancialReports (**

**ReportID INT AUTO\_INCREMENT PRIMARY KEY,**

**Date DATE,**

**TotalIncome DECIMAL(10, 2),**

**TotalTicketsSold INT,**

**UserID INT,**

**FOREIGN KEY (UserID) REFERENCES Users(UserID)**

**);**

**Вставка даних:**

**INSERT INTO Users (Username, Password, Role) VALUES**

**('admin1', 'password1', 'admin'),**

**('admin2', 'password2', 'admin'),**

**('cashier1', 'password3', 'cashier'),**

**('cashier2', 'password4', 'cashier');**

**INSERT INTO Performances (Title, Date, Time, Duration, Description) VALUES**

**('Hamlet', '2023-06-15', '19:00:00', 120, 'A classic Shakespeare play'),**

**('The Nutcracker', '2023-06-20', '18:00:00', 150, 'A famous ballet performance'),**

**('Macbeth', '2023-07-01', '20:00:00', 110, 'Another classic Shakespeare play'),**

**('Swan Lake', '2023-07-10', '19:30:00', 140, 'A well-known ballet performance');**

**INSERT INTO Actors (Name, Bio, Birthdate) VALUES**

**('John Doe', 'Experienced stage actor', '1985-02-15'),**

**('Jane Smith', 'Ballet dancer', '1990-07-22'),**

**('Mike Johnson', 'Shakespearean actor', '1978-11-05'),**

**('Emily Davis', 'Classical dancer', '1992-08-10');**

**INSERT INTO PerformanceActors (PerformanceID, ActorID) VALUES**

**(1, 1), -- John Doe in Hamlet**

**(2, 2), -- Jane Smith in The Nutcracker**

**(1, 3), -- Mike Johnson in Hamlet**

**(3, 3), -- Mike Johnson in Macbeth**

**(4, 4), -- Emily Davis in Swan Lake**

**(2, 4); -- Emily Davis in The Nutcracker**

**INSERT INTO Tickets (PerformanceID, SeatNumber, Price, SaleDate, CustomerName, CustomerContact) VALUES**

**(1, 'A1', 50.00, '2023-05-01 10:00:00', 'Alice Brown', 'alice@example.com'),**

**(1, 'A2', 50.00, '2023-05-01 10:05:00', 'Bob Green', 'bob@example.com'),**

**(2, 'B1', 75.00, '2023-05-02 11:30:00', 'Charlie White', 'charlie@example.com'),**

**(2, 'B2', 75.00, '2023-05-02 11:35:00', 'Dana Blue', 'dana@example.com'),**

**(3, 'C1', 60.00, '2023-05-03 12:00:00', 'Eve Black', 'eve@example.com'),**

**(4, 'D1', 80.00, '2023-05-04 13:00:00', 'Frank Red', 'frank@example.com');**

**INSERT INTO Reservations (PerformanceID, SeatNumber, CustomerName, CustomerContact, ReservationDate, Status) VALUES**

**(1, 'A3', 'Gina Yellow', 'gina@example.com', '2023-05-01 09:00:00', 'reserved'),**

**(2, 'B3', 'Harry Blue', 'harry@example.com', '2023-05-02 12:00:00', 'paid'),**

**(3, 'C2', 'Ivy Pink', 'ivy@example.com', '2023-05-03 11:00:00', 'cancelled'),**

**(4, 'D2', 'Jack Grey', 'jack@example.com', '2023-05-04 14:00:00', 'reserved');**

**INSERT INTO FinancialReports (Date, TotalIncome, TotalTicketsSold, UserID) VALUES**

**('2023-05-01', 125.00, 2, 1),**

**('2023-05-02', 150.00, 2, 2),**

**('2023-05-03', 60.00, 1, 1),**

**('2023-05-04', 80.00, 1, 2);**

**Запити:**

1. **Отримати розклад вистав**

**SELECT Title, Date, Time, Duration, Description**

**FROM Performances;**

1. **Отримати список акторів, які беруть участь у конкретній виставі**

**SELECT A.Name, A.Bio, A.Birthdate**

**FROM Actors A**

**JOIN PerformanceActors PA ON A.ActorID = PA.ActorID**

**WHERE PA.PerformanceID = 1;**

1. **Отримати інформацію про продані квитки для конкретної вистави**

**SELECT TicketID, SeatNumber, Price, SaleDate, CustomerName, CustomerContact**

**FROM Tickets**

**WHERE PerformanceID = 1;**

1. **Отримати інформацію про всі замовлення для конкретної вистави**

**SELECT ReservationID, SeatNumber, CustomerName, CustomerContact, ReservationDate, Status**

**FROM Reservations**

**WHERE PerformanceID = 1;**

1. **Отримати фінансовий звіт за конкретний період**

**SELECT Date, TotalIncome, TotalTicketsSold, U.Username**

**FROM FinancialReports FR**

**JOIN Users U ON FR.UserID = U.UserID**

**WHERE Date BETWEEN '2023-05-01' AND '2023-05-31';**

1. **Отримати загальний дохід за всі вистави**

**SELECT SUM(TotalIncome) AS TotalIncome**

**FROM FinancialReports;**

1. **Отримати загальну кількість проданих квитків за всі вистави**

**SELECT SUM(TotalTicketsSold) AS TotalTicketsSold**

**FROM FinancialReports;**

1. **Отримати середню ціну квитка для конкретної вистави**

**SELECT AVG(Price) AS AverageTicketPrice**

**FROM Tickets**

**WHERE PerformanceID = 1;**

1. **Отримати імена всіх клієнтів, які купили квитки на конкретну виставу**

**SELECT DISTINCT CustomerName**

**FROM Tickets**

**WHERE PerformanceID = 1;**

1. **Отримати всі вистави, на які заброньовані квитки**

**SELECT DISTINCT P.Title**

**FROM Performances P**

**JOIN Reservations R ON P.PerformanceID = R.PerformanceID**

**WHERE R.Status = 'reserved';**

1. **Отримати кількість проданих квитків для кожної вистави**

**SELECT P.Title, COUNT(T.TicketID) AS TicketsSold**

**FROM Performances P**

**JOIN Tickets T ON P.PerformanceID = T.PerformanceID**

**GROUP BY P.Title;**

1. **Отримати список вистав, які проводяться в червні 2023 року**

**SELECT Title, Date, Time, Duration, Description**

**FROM Performances**

**WHERE Date BETWEEN '2023-06-01' AND '2023-06-30';**

1. **Отримати кількість замовлень для кожного статусу**

**SELECT Status, COUNT(ReservationID) AS ReservationCount**

**FROM Reservations**

**GROUP BY Status;**

1. **Отримати імена та контактні дані всіх клієнтів, які зробили бронювання, але не викупили квитки**

**SELECT CustomerName, CustomerContact**

**FROM Reservations**

**WHERE Status = 'reserved';**

1. **Отримати вистави, на які вистачає акторів**

**SELECT P.Title, COUNT(PA.ActorID) AS ActorsCount**

**FROM Performances P**

**JOIN PerformanceActors PA ON P.PerformanceID = PA.PerformanceID**

**GROUP BY P.Title**

**HAVING ActorsCount >= 2;**

1. **Отримати список вистав з їх датами, відсортованих за датою**

**SELECT Title, Date, Time, Duration, Description**

**FROM Performances**

**ORDER BY Date;**

1. **Отримати список акторів, які беруть участь більше ніж в одній виставі**

**SELECT A.Name, COUNT(PA.PerformanceID) AS PerformanceCount**

**FROM Actors A**

**JOIN PerformanceActors PA ON A.ActorID = PA.ActorID**

**GROUP BY A.Name**

**HAVING PerformanceCount > 1;**

1. **Отримати список вистав, на які продані квитки більше ніж на 1000 доларів**

**SELECT P.Title, SUM(T.Price) AS TotalSales**

**FROM Performances P**

**JOIN Tickets T ON P.PerformanceID = T.PerformanceID**

**GROUP BY P.Title**

**HAVING TotalSales > 1000;**

1. **Отримати вистави, на які не було продано жодного квитка**

**SELECT P.Title**

**FROM Performances P**

**LEFT JOIN Tickets T ON P.PerformanceID = T.PerformanceID**

**WHERE T.TicketID IS NULL;**

1. **Отримати деталі всіх квитків, проданих конкретним касиром**

**SELECT T.TicketID, T.SeatNumber, T.Price, T.SaleDate, T.CustomerName, T.CustomerContact**

**FROM Tickets T**

**JOIN FinancialReports FR ON DATE(T.SaleDate) = FR.Date**

**JOIN Users U ON FR.UserID = U.UserID**

**WHERE U.UserID = 2;**

1. **Отримати сумарний дохід за всі вистави для кожного касира**

**SELECT U.Username, SUM(FR.TotalIncome) AS TotalIncome**

**FROM Users U**

**JOIN FinancialReports FR ON U.UserID = FR.UserID**

**GROUP BY U.Username;**

1. **Отримати кількість вистав, які кожен актор відвідує в червні 2023 року**

**SELECT A.Name, COUNT(PA.PerformanceID) AS PerformanceCount**

**FROM Actors A**

**JOIN PerformanceActors PA ON A.ActorID = PA.ActorID**

**JOIN Performances P ON PA.PerformanceID = P.PerformanceID**

**WHERE P.Date BETWEEN '2023-06-01' AND '2023-06-30'**

**GROUP BY A.Name;**

1. **Отримати найвищу та найнижчу ціну квитка для конкретної вистави**

**SELECT MAX(Price) AS MaxPrice, MIN(Price) AS MinPrice**

**FROM Tickets**

**WHERE PerformanceID = 1;**

1. **Отримати середню кількість проданих квитків на виставу**

**SELECT AVG(TicketsSold) AS AverageTicketsSold**

**FROM (**

**SELECT PerformanceID, COUNT(TicketID) AS TicketsSold**

**FROM Tickets**

**GROUP BY PerformanceID**

**) AS SubQuery;**

1. **Отримати список клієнтів, які купили квитки більше ніж на одну виставу**

**SELECT CustomerName, COUNT(DISTINCT PerformanceID) AS PerformancesCount**

**FROM Tickets**

**GROUP BY CustomerName**

**HAVING PerformancesCount > 1;**

1. **Отримати сумарний дохід та кількість проданих квитків за кожний день**

**SELECT Date(SaleDate) AS SaleDay, SUM(Price) AS TotalIncome, COUNT(TicketID) AS TotalTicketsSold**

**FROM Tickets**

**GROUP BY SaleDay;**

1. **Отримати список клієнтів, які купили квитки на вистави більше ніж на 100 доларів**

**SELECT CustomerName, SUM(Price) AS TotalSpent**

**FROM Tickets**

**GROUP BY CustomerName**

**HAVING TotalSpent > 100;**

1. **Отримати кількість вистав, на які було зроблено хоча б одне замовлення**

**SELECT COUNT(DISTINCT PerformanceID) AS PerformancesWithReservations**

**FROM Reservations;**

1. **Отримати всі вистави, які відвідує конкретний актор**

**SELECT P.Title, P.Date, P.Time, P.Duration, P.Description**

**FROM Performances P**

**JOIN PerformanceActors PA ON P.PerformanceID = PA.PerformanceID**

**WHERE PA.ActorID = 1;**

1. **Отримати список акторів, які не беруть участь у жодній виставі**

**SELECT A.Name, A.Bio, A.Birthdate**

**FROM Actors A**

**LEFT JOIN PerformanceActors PA ON A.ActorID = PA.ActorID**

**WHERE PA.PerformanceID IS NULL;**

1. **Отримати список клієнтів з контактною інформацією, які зробили бронювання в конкретний період**

**SELECT CustomerName, CustomerContact**

**FROM Reservations**

**WHERE ReservationDate BETWEEN '2023-06-01' AND '2023-06-30';**

1. **Отримати середній дохід від продажу квитків на кожну виставу**

**SELECT P.Title, AVG(T.Price) AS AverageIncome**

**FROM Performances P**

**JOIN Tickets T ON P.PerformanceID = T.PerformanceID**

**GROUP BY P.Title;**

1. **Отримати список вистав, які були відвідані більш ніж 5 різними клієнтами**

**SELECT P.Title, COUNT(DISTINCT T.CustomerName) AS UniqueCustomers**

**FROM Performances P**

**JOIN Tickets T ON P.PerformanceID = T.PerformanceID**

**GROUP BY P.Title**

**HAVING UniqueCustomers > 5;**

1. **Отримати сумарний дохід за всі вистави для кожного місяця**

**SELECT DATE\_FORMAT(SaleDate, '%Y-%m') AS SaleMonth, SUM(Price) AS TotalIncome**

**FROM Tickets**

**GROUP BY SaleMonth;**

1. **Отримати сумарний дохід та кількість проданих квитків за кожен тиждень**

**SELECT YEAR(SaleDate) AS Year, WEEK(SaleDate) AS Week, SUM(Price) AS TotalIncome, COUNT(TicketID) AS TotalTicketsSold**

**FROM Tickets**

**GROUP BY Year, Week;**