**Cấu trúc Project CSDL Lab**

**I. Phân tích yêu cầu (Requirement Analysis)**

1. **Các chức năng chính (Core Features)**
   * **Quản lý Phim/Sự kiện**
     + Thêm, sửa, xoá thông tin phim: tiêu đề, thể loại, thời lượng, quốc gia sản xuất, mô tả, hình ảnh bìa.
     + Lên lịch suất chiếu: ngày, giờ, rạp, giá vé.
     + Cập nhật trạng thái (đang chiếu, đã ngừng chiếu).
   * **Quản lý Rạp và Ghế**
     + Thông tin rạp: mã rạp, tên rạp, số lượng hàng/ghế, cấu hình phòng.
     + Thông tin ghế: mã ghế, vị trí (hàng, cột), trạng thái (hỗ trợ tốt, hỗ trợ cho khách khuyết tật, v.v.).
   * **Quản lý Khách hàng**
     + Đăng ký/Đăng nhập (tên tài khoản, mật khẩu, email, số điện thoại).
     + Thông tin cá nhân (họ tên, ngày sinh, giới tính, địa chỉ).
     + Lịch sử đặt vé, điểm tích luỹ (nếu có).
   * **Quản lý Nhân viên**
     + Thông tin nhân viên (mã NV, họ tên, email, vai trò, rạp phụ trách).
     + Phân quyền (đăng nhập, xem/sửa phim, phê duyệt đơn đặt).
   * **Quản lý Đơn hàng (Order) & Chi tiết Đơn hàng (OrderDetail)**
     + Khi khách chọn suất chiếu, chọn ghế → tạo đơn đặt vé.
     + Lưu trạng thái (Chưa thanh toán, Đã thanh toán, Đã hủy).
     + Chi tiết vé: mã ghế, giá vé, mã suất chiếu, số lượng, tổng tiền.
   * **Quản lý Thanh toán và In vé**
     + Xác thực thanh toán (online/offline), tạo mã vé điện tử (QR code/số ghế).
     + Lưu thông tin phương thức (thẻ tín dụng, ví điện tử, tiền mặt).
   * **Thống kê – Báo cáo**
     + Doanh thu theo ngày/tháng/quý/năm.
     + Top phim/ suất chiếu/ rạp bán chạy.
     + Báo cáo bán vé theo nhân viên.
   * **Phân quyền & Bảo mật**
     + Ba nhóm vai trò chính:
       1. **Customer** (khách hàng)
       2. **Employee** (nhân viên bán vé)
       3. **Admin/Manager** (quản trị hệ thống/rạp)
     + Quyền hạn:
       1. Customer: Đặt vé, xem lịch sử, huỷ vé (trong thời gian cho phép).
       2. Employee: Quản lý phim, suất chiếu, xử lý đơn đặt, xem báo cáo cơ bản.
       3. Admin: Quản lý toàn bộ, phân quyền nhân viên, xem báo cáo tổng quan, tạo tài khoản.
   * **Các yêu cầu không chức năng (Non-Functional)**
     + **Tính toàn vẹn dữ liệu**: Sử dụng ràng buộc khóa ngoại, kiểm tra trạng thái ghế.
     + **Hiệu suất**: Chỉ số Mục lục trên cột thường xuyên tìm kiếm (ngày chiếu, rạp, phim).
     + **Bảo mật**: Mật khẩu mã hóa, phân quyền truy cập (role-based).
2. **Use Cases / Scenarios tiêu biểu**
   * **Khách hàng A đặt vé “Avengers: Endgame”**
     + Actor: Customer
     + Bước:
       1. Chọn phim → chọn suất chiếu (ngày 10/06/2025, 19:00, rạp 5)
       2. Hệ thống hiển thị sơ đồ chỗ còn trống → Chọn ghế D5, D6
       3. Xác nhận thông tin cá nhân → Thanh toán
       4. Hệ thống tạo Order + OrderDetail, gửi email/xuất mã vé
     + Kết quả: Vé được ghi nhận, khách hàng có mã giao dịch.
   * **Nhân viên B thêm phim mới**
     + Actor: Employee
     + Bước:
       1. Đăng nhập → Chọn “Quản lý Phim” → “Thêm Phim”
       2. Nhập thông tin Phim: “Spiderman: No Way Home”, Hành động, 2h, 5 suất/ngày, giá vé 120.000₫/vé.
       3. Lên lịch suất chiếu cho 5 ngày tiếp theo, lưu rạp, ngày/giờ cụ thể.
     + Kết quả: Phim mới và lịch suất chiếu được ghi nhận.
   * **Admin xem báo cáo doanh thu tháng 06/2025**
     + Actor: Admin
     + Bước:
       1. Đăng nhập → Chọn “Báo Cáo” → “Doanh thu theo Tháng”
       2. Nhập Tháng = 06/2025
       3. Hệ thống hiển thị tổng doanh thu, phân rạp, phân phim, top 5 suất chiếu.
     + Kết quả: Báo cáo được hiển thị, Admin có thể xuất file Excel/PDF.
3. **Phân quyền (User Roles & Permissions)**
   * **Customer**
     + Quyền:
       1. Xem danh sách phim/ suất chiếu đang có
       2. Đặt vé (chọn ghế, thanh toán)
       3. Xem/huỷ lịch sử vé đã đặt
   * **Employee**
     + Quyền:
       1. Quản lý Phim (Thêm/Sửa/Xóa)
       2. Quản lý Suất chiếu (Thêm/Sửa/Xóa)
       3. Xem & xử lý Đơn hàng (phê duyệt, huỷ)
       4. Quản lý Ghế (set trạng thái cố định/hỏng)
       5. Xem báo cáo cơ bản (doanh thu theo ngày, top phim)
   * **Admin/Manager**
     + Quyền:
       1. Tất cả quyền của Employee
       2. Quản lý Nhân viên (Tạo tài khoản, gán role, điều chỉnh quyền)
       3. Xem báo cáo nâng cao (theo rạp, theo nhân viên, theo khu vực)
       4. Backup/Restore database, cấu hình hệ thống

**II. Thiết kế Cơ sở dữ liệu (Logical/Data Modeling)**

**2.1. Danh sách Bảng (Thực thể)**

Dưới đây là danh sách các bảng chính (có thể điều chỉnh thêm/bớt).

**2.1.1. Bảng Movie (Phim)**

* **movie\_id** (INT, PK, IDENTITY)
* **title** (NVARCHAR(200), NOT NULL)
* **genre** (NVARCHAR(100))
* **duration\_minutes** (INT)
* **language** (NVARCHAR(50))
* **description** (NVARCHAR(MAX), NULL)
* **status** (CHAR(10), CHECK IN (‘ĐangChiếu’, ‘NgừngChiếu’), DEFAULT ‘ĐangChiếu’)
* **poster\_url** (NVARCHAR(500), NULL)
* **created\_at** (DATETIME, DEFAULT GETDATE())
* **updated\_at** (DATETIME, DEFAULT GETDATE())

**2.1.2. Bảng Cinema (Rạp)**

* **cinema\_id** (INT, PK, IDENTITY)
* **name** (NVARCHAR(100), NOT NULL)
* **location** (NVARCHAR(200), NULL)
* **total\_rows** (INT, NOT NULL)
* **total\_columns** (INT, NOT NULL)
* **created\_at** (DATETIME, DEFAULT GETDATE())

**2.1.3. Bảng Seat (Ghế)**

* **seat\_id** (INT, PK, IDENTITY)
* **cinema\_id** (INT, FK → Cinema(cinema\_id), NOT NULL)
* **row\_number** (NVARCHAR(5), NOT NULL)
* **col\_number** (INT, NOT NULL)
* **seat\_type** (NVARCHAR(50), CHECK IN (‘Thường’, ‘VIP’, ‘NgườiKhuyếtTật’), DEFAULT ‘Thường’)
* **status** (CHAR(10), CHECK IN (‘Available’, ‘Unavailable’), DEFAULT ‘Available’)

**Chú ý**: Ta có thể gán mỗi ghế cho một rạp, để tái sử dụng khi tạo các phòng chiếu riêng cho từng suất.

**2.1.4. Bảng Showtime (Suất chiếu)**

* **showtime\_id** (INT, PK, IDENTITY)
* **movie\_id** (INT, FK → Movie(movie\_id), NOT NULL)
* **cinema\_id** (INT, FK → Cinema(cinema\_id), NOT NULL)
* **start\_time** (DATETIME, NOT NULL)
* **price** (DECIMAL(10,2), NOT NULL)
* **total\_seats** (INT, NOT NULL) – tổng số ghế của phòng, nhân viên không cần nhập nếu lấy tự động từ Cinema
* **available\_seats** (INT, NOT NULL) – cập nhật khi có đặt/huỷ vé
* **created\_at** (DATETIME, DEFAULT GETDATE())

**2.1.5. Bảng Customer (Khách hàng)**

* **customer\_id** (INT, PK, IDENTITY)
* **username** (NVARCHAR(50), UNIQUE, NOT NULL)
* **password\_hash** (NVARCHAR(200), NOT NULL)
* **full\_name** (NVARCHAR(100), NOT NULL)
* **email** (NVARCHAR(150), UNIQUE, NOT NULL)
* **phone** (NVARCHAR(20), NULL)
* **dob** (DATE, NULL)
* **created\_at** (DATETIME, DEFAULT GETDATE())
* **updated\_at** (DATETIME, DEFAULT GETDATE())

**2.1.6. Bảng Employee (Nhân viên)**

* **employee\_id** (INT, PK, IDENTITY)
* **username** (NVARCHAR(50), UNIQUE, NOT NULL)
* **password\_hash** (NVARCHAR(200), NOT NULL)
* **full\_name** (NVARCHAR(100), NOT NULL)
* **email** (NVARCHAR(150), UNIQUE, NOT NULL)
* **phone** (NVARCHAR(20), NULL)
* **role** (CHAR(10), CHECK IN (‘Employee’, ‘Admin’), DEFAULT ‘Employee’)
* **cinema\_id** (INT, FK → Cinema(cinema\_id), NULL) – nhân viên được phân rạp nếu muốn
* **created\_at** (DATETIME, DEFAULT GETDATE())

**2.1.7. Bảng Order (Đơn hàng)**

* **order\_id** (INT, PK, IDENTITY)
* **customer\_id** (INT, FK → Customer(customer\_id), NOT NULL)
* **employee\_id** (INT, FK → Employee(employee\_id), NULL) – nhân viên xử lý (nếu order offline)
* **order\_date** (DATETIME, DEFAULT GETDATE())
* **total\_amount** (DECIMAL(12,2), NOT NULL)
* **status** (CHAR(20), CHECK IN (‘ChưaThanhToán’, ‘ĐãThanhToán’, ‘ĐãHuỷ’), DEFAULT ‘ChưaThanhToán’)
* **payment\_method** (CHAR(20), CHECK IN (‘Thẻ’, ‘Ví’, ‘TiềnMặt’), NULL)
* **payment\_date** (DATETIME, NULL)

**2.1.8. Bảng OrderDetail (Chi tiết đơn hàng)**

* **order\_detail\_id** (INT, PK, IDENTITY)
* **order\_id** (INT, FK → Order(order\_id), NOT NULL)
* **showtime\_id** (INT, FK → Showtime(showtime\_id), NOT NULL)
* **seat\_id** (INT, FK → Seat(seat\_id), NOT NULL)
* **price** (DECIMAL(10,2), NOT NULL)
* **created\_at** (DATETIME, DEFAULT GETDATE())

**Lưu ý**: Mỗi bản ghi là một ghế khách đặt trong 1 suất chiếu. Giá ở OrderDetail để dự trữ giá tại thời điểm đặt vé (tránh thay đổi khi giá suất chiếu chỉnh sửa sau đó).

**2.1.9. Bảng Payment (Thanh toán) (tùy chọn, nếu muốn tách riêng)**

* **payment\_id** (INT, PK, IDENTITY)
* **order\_id** (INT, FK → Order(order\_id), UNIQUE, NOT NULL)
* **payment\_date** (DATETIME, DEFAULT GETDATE())
* **amount** (DECIMAL(12,2), NOT NULL)
* **method** (CHAR(20), CHECK IN (‘Thẻ’, ‘Ví’, ‘TiềnMặt’), NOT NULL)
* **transaction\_code** (NVARCHAR(100), NULL)

**2.1.10. Bảng Role (Vai trò - tùy chọn nếu muốn quản lý tách riêng)**

* **role\_id** (INT, PK, IDENTITY)
* **role\_name** (CHAR(20), UNIQUE, CHECK IN (‘Customer’, ‘Employee’, ‘Admin’))
* **description** (NVARCHAR(200), NULL)

Nếu bạn muốn mở rộng quyền linh hoạt hơn (RBAC), có thể tạo bảng Role và bảng mapping User–Role. Nhưng trong phạm vi nhỏ, có thể gán role trực tiếp trong bảng Employee/Customer.

**2.1.11. Bảng AuditLog (ghi nhận hoạt động - tùy chọn)**

* **log\_id** (INT, PK, IDENTITY)
* **user\_id** (INT, NULL)
* **user\_type** (CHAR(10), CHECK IN (‘Customer’, ‘Employee’, ‘Admin’))
* **action** (NVARCHAR(200))
* **object\_type** (NVARCHAR(50))
* **object\_id** (INT, NULL)
* **log\_time** (DATETIME, DEFAULT GETDATE())

Giúp theo dõi các hoạt động quan trọng (thêm phim, huỷ vé, đăng nhập thất bại...).

**2.2. Mô hình ER và Quan hệ (Relationships)**

* **Movie (1) – (N) Showtime**
  + Một phim có thể có nhiều suất chiếu → showtime.movie\_id FK → movie.movie\_id.
* **Cinema (1) – (N) Showtime**
  + Một rạp có thể có nhiều suất chiếu → showtime.cinema\_id FK → cinema.cinema\_id.
* **Cinema (1) – (N) Seat**
  + Mỗi rạp có tập hợp ghế → seat.cinema\_id FK → cinema.cinema\_id.
* **Customer (1) – (N) Order**
  + Một khách có thể đặt nhiều đơn → order.customer\_id FK → customer.customer\_id.
* **Employee (1) – (N) Order**
  + Một nhân viên có thể xử lý nhiều đơn offline → order.employee\_id FK → employee.employee\_id.
* **Order (1) – (N) OrderDetail**
  + Một đơn có nhiều chi tiết → order\_detail.order\_id FK → order.order\_id.
* **Showtime (1) – (N) OrderDetail**
  + Một suất chiếu có thể xuất hiện trong nhiều chi tiết đơn → order\_detail.showtime\_id FK → showtime.showtime\_id.
* **Seat (1) – (N) OrderDetail**
  + Mỗi ghế được đặt tối đa một lần cho một suất (kiểm soát ở ứng dụng/trigger) → order\_detail.seat\_id FK → seat.seat\_id.

**Chú ý**: Khi tạo order, phải kiểm tra xem ghế có còn trống (chưa có trong OrderDetail cho suất đó).

**III. Danh sách Stored Procedures (Procedures & Functions)**

Dưới đây là một số procedure/phương thức cần có. Bạn có thể thêm/sửa tùy theo nghiệp vụ:

1. **sp\_AddMovie**
   * Thêm phim mới (có thể kèm tải hình, mô tả).
   * Input: title, genre, duration, language, description, poster\_url.
   * Output: movie\_id được tạo.
2. **sp\_UpdateMovie**
   * Cập nhật thông tin phim.
   * Input: movie\_id, [các trường cần cập nhật].
3. **sp\_DeleteMovie**
   * Xoá phim (hoặc đánh dấu status = ‘NgừngChiếu’).
   * Input: movie\_id.
4. **sp\_AddShowtime**
   * Thêm suất chiếu mới.
   * Input: movie\_id, cinema\_id, start\_time, price.
   * Tính total\_seats = số ghế của rạp; available\_seats = total\_seats.
5. **sp\_UpdateShowtime**
   * Cập nhật suất chiếu.
   * Input: showtime\_id, [các trường có thể sửa].
6. **sp\_DeleteShowtime**
   * Xoá suất chiếu (hoặc đặt trạng thái ngừng).
   * Input: showtime\_id.
7. **sp\_GetAvailableSeats**
   * Lấy danh sách ghế trống cho suất chiếu.
   * Input: showtime\_id
   * Logic:

SELECT s.seat\_id, s.row\_number, s.col\_number

FROM Seat s

JOIN Cinema c ON s.cinema\_id = c.cinema\_id

WHERE s.cinema\_id = (SELECT cinema\_id FROM Showtime WHERE showtime\_id = @showtime\_id)

AND s.seat\_id NOT IN (

SELECT od.seat\_id

FROM OrderDetail od

JOIN [Order] o ON od.order\_id = o.order\_id

WHERE od.showtime\_id = @showtime\_id

AND o.status = 'ĐãThanhToán'

)

AND s.status = 'Available';

1. **sp\_CreateOrder**
   * Tạo đơn đặt vé (transaction):
     1. Kiểm tra còn đủ ghế.
     2. Tạo bản ghi Order (status = ‘ChưaThanhToán’).
     3. Thêm từng OrderDetail (mỗi ghế).
     4. Giảm available\_seats trong Showtime.
   * Input: customer\_id, danh sách @list\_seat\_id, showtime\_id, payment\_method, employee\_id (nếu cần).
   * Output: order\_id, tổng tiền.
2. **sp\_PayOrder**
   * Cập nhật trạng thái Order → ‘ĐãThanhToán’, cập nhật Payment.
   * Input: order\_id, payment\_method, transaction\_code (nếu có).
   * Logic:

UPDATE [Order]

SET status = 'ĐãThanhToán', payment\_method = @payment\_method, payment\_date = GETDATE()

WHERE order\_id = @order\_id;

1. **sp\_CancelOrder**
   * Huỷ đơn (trong thời gian cho phép):
     1. Input: order\_id.
     2. Logic:
        1. Kiểm tra trạng thái hiện tại (ChưaThanhToán hoặc ĐãThanhToán theo chính sách).
        2. Cập nhật Order.status = 'ĐãHuỷ'.
        3. Xoá/bỏ khóa ghế → tăng available\_seats.
        4. (Nếu đã thanh toán) Xử lý hoàn tiền (nếu có).
2. **sp\_GetMovieSchedule**
   * Lấy lịch chiếu (theo phim, theo ngày, theo rạp).
   * Input: @movie\_id (NULL nếu lấy tất cả), @date\_from, @date\_to, @cinema\_id (NULL).
   * Trả về: danh sách showtime với số ghế trống.
3. **sp\_GetRevenueByDate**
   * Lấy doanh thu theo ngày.
   * Input: @start\_date, @end\_date.
   * Output: ngày, tổng tiền, số lượng vé bán, số đơn.
4. **sp\_GetTopMovies**
   * Lấy top n phim bán chạy (theo số vé hoặc doanh thu) trong khoảng thời gian.
   * Input: @start\_date, @end\_date, @top\_n.
   * Logic:

SELECT m.movie\_id, m.title, COUNT(od.order\_detail\_id) AS tickets\_sold,

SUM(od.price) AS revenue

FROM [OrderDetail] od

JOIN [Order] o ON od.order\_id = o.order\_id AND o.status = 'ĐãThanhToán'

JOIN Showtime st ON od.showtime\_id = st.showtime\_id

JOIN Movie m ON st.movie\_id = m.movie\_id

WHERE o.order\_date BETWEEN @start\_date AND @end\_date

GROUP BY m.movie\_id, m.title

ORDER BY revenue DESC;

1. **sp\_GetTopCinemas**
   * Tương tự sp\_GetTopMovies nhưng theo rạp.
2. **sp\_AddCustomer**
   * Đăng ký khách hàng mới, kiểm tra trùng username/email.
   * Input: username, password\_hash, full\_name, email, phone, dob.
3. **sp\_Login**
   * Xác thực người dùng chung (customer hoặc employee).
   * Input: username, password\_hash.
   * Output: user\_id, role.
   * (Có thể tách riêng sp\_CustomerLogin và sp\_EmployeeLogin nếu muốn phân chia rõ ràng).
4. **sp\_AddEmployee**, **sp\_UpdateEmployee**, **sp\_DeleteEmployee**
   * Tương tự quản lý Customer nhưng dành cho nhân viên (có role, gán rạp).

**IV. Views (nếu cần)**

1. **vw\_AvailableShowtimes**
   * Hiển thị danh sách suất chiếu có ghế trống (có thể search theo ngày, phim, rạp):

CREATE VIEW vw\_AvailableShowtimes AS

SELECT st.showtime\_id, m.title, c.name AS cinema\_name,

st.start\_time, st.price, st.available\_seats

FROM Showtime st

JOIN Movie m ON st.movie\_id = m.movie\_id

JOIN Cinema c ON st.cinema\_id = c.cinema\_id

WHERE st.available\_seats > 0;

1. **vw\_CinemaSeatStatus**
   * Cho biết trạng thái toàn bộ ghế theo rạp, theo suất:

CREATE VIEW vw\_CinemaSeatStatus AS

SELECT st.showtime\_id, c.name AS cinema\_name, s.row\_number, s.col\_number,

CASE

WHEN s.seat\_id IN (

SELECT od.seat\_id

FROM OrderDetail od

JOIN [Order] o ON od.order\_id = o.order\_id

WHERE od.showtime\_id = st.showtime\_id AND o.status = 'ĐãThanhToán'

) THEN 'Booked'

ELSE 'Available'

END AS seat\_status

FROM Seat s

CROSS JOIN Showtime st

WHERE s.cinema\_id = st.cinema\_id;

1. **vw\_DailyRevenue**
   * Doanh thu, số vé bán theo ngày trong tháng:

CREATE VIEW vw\_DailyRevenue AS

SELECT CAST(o.order\_date AS DATE) AS [Date],

SUM(od.price) AS TotalRevenue,

COUNT(od.order\_detail\_id) AS TicketsSold

FROM [Order] o

JOIN OrderDetail od ON o.order\_id = od.order\_id

WHERE o.status = 'ĐãThanhToán'

GROUP BY CAST(o.order\_date AS DATE);

1. **vw\_MonthlyRevenue**
   * Tổng doanh thu theo tháng, năm:

CREATE VIEW vw\_MonthlyRevenue AS

SELECT YEAR(o.order\_date) AS [Year], MONTH(o.order\_date) AS [Month],

SUM(od.price) AS TotalRevenue, COUNT(od.order\_detail\_id) AS TicketsSold

FROM [Order] o

JOIN OrderDetail od ON o.order\_id = od.order\_id

WHERE o.status = 'ĐãThanhToán'

GROUP BY YEAR(o.order\_date), MONTH(o.order\_date);

**V. Phân quyền (Roles & Permissions)**

Nếu sử dụng tính năng **SQL Server Role-Based Security**, bạn có thể:

1. **Tạo Server Roles/ Database Roles**

-- Tạo database role cho Customer

CREATE ROLE db\_Customer;

-- Tạo database role cho Employee

CREATE ROLE db\_Employee;

-- Tạo database role cho Admin

CREATE ROLE db\_Admin;

1. **Gán Permissions**
   * **db\_Customer**

GRANT SELECT ON vw\_AvailableShowtimes TO db\_Customer;

EXEC sp\_addrolemember 'db\_Customer', 'Customer\_UserName';

* + **db\_Employee**

GRANT SELECT, INSERT, UPDATE ON Movie TO db\_Employee;

GRANT SELECT, INSERT, UPDATE ON Showtime TO db\_Employee;

GRANT EXECUTE ON sp\_GetAvailableSeats TO db\_Employee;

GRANT EXECUTE ON sp\_CreateOrder TO db\_Employee;

…

EXEC sp\_addrolemember 'db\_Employee', 'Employee\_UserName';

* + **db\_Admin**

GRANT CONTROL ON DATABASE::[YourDatabaseName] TO db\_Admin;

-- Hoặc gán tất cả quyền (SELECT/INSERT/UPDATE/DELETE/EXECUTE) lên tất cả schema

EXEC sp\_addrolemember 'db\_Admin', 'Admin\_UserName';

1. **Mapping User → Role**
   * Khi tạo login/user, gán vào các role tương ứng dựa vào bảng Employee/Customer.
   * Có thể tự động hóa qua trigger hoặc build trong ứng dụng (VB.NET, C#, Java, v.v.) khi đăng ký tài khoản.

**VI. Cấu trúc các Procedures và Queries Quan trọng**

* Quản lý nghiệp vụ đặt vé (CreateOrder/PayOrder/CancelOrder)
* Xem lịch chiếu và chọn ghế (GetAvailableSeats/GetMovieSchedule)
* Quản lý dữ liệu tham số (Add/Update/Delete Movie, Showtime, Cinema, Seat)
* Thống kê – Báo cáo (GetRevenueByDate, GetTopMovies, GetTopCinemas)
* Xác thực & phân quyền (Login, phân quyền trực tiếp hoặc qua Role)

**VII. Cấu trúc Thư mục (Folder Structure)**

Trong project (dự án SQL), bạn nên chia các script thành từng thư mục rõ ràng. Ví dụ:

/SQLProject

│

├── 01\_Schemas/

│ ├── 01\_CinemaTable.sql

│ ├── 02\_SeatTable.sql

│ ├── 03\_MovieTable.sql

│ ├── 04\_ShowtimeTable.sql

│ ├── 05\_CustomerTable.sql

│ ├── 06\_EmployeeTable.sql

│ ├── 07\_OrderTable.sql

│ ├── 08\_OrderDetailTable.sql

│

├── 02\_Views/

│ ├── vw\_AvailableShowtimes.sql

│ ├── vw\_CinemaSeatStatus.sql

│ ├── vw\_DailyRevenue.sql

│ └── vw\_MonthlyRevenue.sql

│

├── 03\_StoredProcedures/

│ ├── sp\_AddMovie.sql

│ ├── sp\_UpdateMovie.sql

│ ├── sp\_DeleteMovie.sql

│ ├── sp\_AddShowtime.sql

│ ├── sp\_UpdateShowtime.sql

│ ├── sp\_DeleteShowtime.sql

│ ├── sp\_GetAvailableSeats.sql

│ ├── sp\_CreateOrder.sql

│ ├── sp\_PayOrder.sql

│ ├── sp\_CancelOrder.sql

│ ├── sp\_GetMovieSchedule.sql

│ ├── sp\_GetRevenueByDate.sql

│ ├── sp\_GetRevenueByMonth.sql

│ ├── sp\_GetTopMovies.sql

│ ├── sp\_GetTopCinemas.sql

│ ├── sp\_AddCustomer.sql

│ ├── sp\_Login.sql

│ ├── sp\_AddEmployee.sql

│ ├── sp\_UpdateEmployee.sql

│ └── sp\_DeleteEmployee.sql

│

├── 04\_Roles\_Permissions/

│ ├── 00\_CreateLogins.sql

│ ├── 01\_CreateUsers.sql

│ ├── 02\_CreateDBRoles.sql

│ ├── 03\_GrantPermissions.sql

│ └── GrantPermissions\_Customer.sql

│ └── GrantPermissions\_Employee.sql

│ └── GrantPermissions\_Admin.sql

│ └── 04\_RevokePermissions.sql

│ └── RevokePermissions\_Customer.sql

│ └── RevokePermissions\_Employee.sql

│ └── RevokePermissions\_Admin.sql

│

├── 05\_TestScripts/ (script để test từng stored procedure / view)

│ ├── TestScript1\_CreateShema.sql

│ ├── TestScript2\_Customer.sql

│ ├── TestScript3\_Employe.sql

│ ├── TestScript4\_Admin.sql

│ └── TestScript5\_RolePermission.sql

│

└── README.md (Hướng dẫn chạy, thứ tự import các script, mô tả project)

* **01\_Schemas/**: Chứa script tạo các bảng (theo thứ tự phụ thuộc khóa ngoại).
* **02\_Views/**: Lưu tất cả các view đã định nghĩa.
* **03\_StoredProcedures/**: Lưu trữ tất cả procedure/function.
* **04\_Roles\_Permissions/**: Script khởi tạo database roles và gán quyền.
* **05\_TestScripts/**: Script dùng để gọi thử và kiểm tra kết quả cho từng procedure/view.
* **README.md**: Giải thích tổng quan, thứ tự chạy, đặc tả các đối tượng, cách cài đặt, yêu cầu server (SQL Server version…), v.v.

**VIII. Lưu ý khi triển khai**

1. **Thứ tự thực hiện các script**
   * Tạo Database → Tạo các bảng (Schemas) → Seed dữ liệu (nếu cần) → Tạo Views → Tạo Stored Procedures → Tạo Roles & Gán Quyền → Chạy TestScripts.
2. **Transaction và Kiểm tra Tính toàn vẹn**
   * Với các procedure nhạy cảm (sp\_CreateOrder, sp\_CancelOrder), sử dụng BEGIN TRAN / COMMIT / ROLLBACK để đảm bảo atomicity.
   * Kiểm tra ràng buộc (Constraint) trên cột, ví dụ:

ALTER TABLE OrderDetail

ADD CONSTRAINT UQ\_OrderShowtimeSeat UNIQUE (showtime\_id, seat\_id);

– Đảm bảo cùng ghế không được đặt 2 lần cho một suất.

1. **Indexing**
   * Khuyến nghị tạo index trên các cột thường tìm kiếm lọc:
     + Showtime(start\_time)
     + Order(order\_date), Order(status)
     + Movie(status)
     + OrderDetail(showtime\_id)
2. **Backup & Phục hồi**
   * Định kỳ backup database, nhất là giai đoạn gần triển khai.
   * Trong folder có thể có script backup (tùy server), hoặc hướng dẫn (README).
3. **Test và Kiểm thử**
   * Mỗi lập trình viên trong nhóm nên được assign một phần (ví dụ: một người thiết kế schema + seed data, một người viết procedure + view, một người viết script test + phân quyền).
   * Sau khi hoàn thành, chạy toàn bộ từ đầu để đảm bảo không phát sinh lỗi.

**IX. Tóm tắt**

* **Các bảng chính**: Movie, Cinema, Seat, Showtime, Customer, Employee, Order, OrderDetail (có thể thêm Payment, Role, AuditLog nếu cần).
* **Quan hệ**: Khóa ngoại giữa Movie–Showtime, Cinema–Showtime, Cinema–Seat, Customer–Order, Employee–Order, Order–OrderDetail, Showtime–OrderDetail, Seat–OrderDetail.
* **Stored Procedures**: Thực hiện thêm/sửa/xoá phim, suất chiếu; tạo/huỷ/ thanh toán đơn hàng; thống kê doanh thu; đăng ký khách/nhân viên; xác thực người dùng.
* **Views**: Hỗ trợ hiển thị suất chiếu, trạng thái ghế, báo cáo doanh thu theo ngày/tháng.
* **Phân quyền**: 3 role (Customer, Employee, Admin) với quyền hạn từ cơ bản đến toàn quyền.
* **Cấu trúc thư mục**: Chia rõ Scripts theo Schema, Views, StoredProcedures, Roles\_Permissions, SeedData, TestScripts, kèm README.md.