

EECS3421 GROUP PROJECT

Final Report

Case: [Cineplex](#) (9)
Course Section: B
Team Number: Group 9

Student Name	Student ID	Email
Mohammad Areeb	220094736	areeb157@my.yorku.ca
Prabhkrit Singh	220164059	krit@my.yorku.ca
Akash Deep	220854162	akash7@my.yorku.ca
Karthikeya Gorijavolu	220241030	gkarthik@my.yorku.ca
Samarjeet Singh Arora	220857686	samar15@my.yorku.ca

Introduction and Preliminary Study

The Organization

Cineplex is an actual organization, which is amongst the most significant entertainment organizations in Canada itself as it operates several cinemas in the country, as well as offering entertainment services such as film screening facilities, VIP cinemas, game zones (Playdium), as well as event hosting for sports, concerts, etc. Cineplex also has an online platform where customers get to watch programming from the comfort of their homes. The firm was established in 1912, with headquarters based in Toronto, Ontario. For many years, this company has grown to be a market leader in the Canadian entertainment industry serving millions of clients every year via physical and digital services.

Business Objective

The primary objective of this database project is to design and implement a comprehensive relational database system that supports the core operations of a modern cinema theatre chain. This system addresses the critical business requirements of movie exhibition, ticket booking, concession sales, and customer loyalty management in an integrated digital platform.

The database aims to enable seamless end-to-end customer experiences by managing the complete lifecycle of movie-going activities, from browsing currently playing films and selecting showtimes to reserving seats, processing multi-method payments, ordering concessions, and participating in loyalty reward programs. By consolidating these diverse business functions into a unified data architecture, the system provides theatre operators with real-time inventory control, revenue tracking, and customer relationship management capabilities.

Key objectives include maintaining data integrity across concurrent booking operations through sophisticated seat locking mechanisms, supporting dynamic pricing strategies based on seat categories and membership benefits, facilitating complex payment workflows that combine credit cards, gift cards, and loyalty points, and providing comprehensive transaction history for both customers and theatre management. The database is designed to scale across multiple theatre locations while maintaining consistent business rules and operational standards.

Scope of the Application

1. Movies and metadata:

- ❖ Maintain movie records with title, synopsis, duration, language, release date, PG rating, and IMDB rating; attach posters/trailers via media files/URLs and upload timestamps.
- ❖ Support many-to-many genres and cast members, including actor roles per movie for accurate discovery and crediting.

2. Venues, screens, and seats:

- ❖ Model multiple theatres with address and contact details; each theatre houses multiple screens with capacity and premium features (VIP, D-Box flags).
- ❖ Store individual seats with row/number and link to seat categories that carry accessibility flags and price modifiers.

3. Showtimes and programming:

- ❖ Schedule showtimes per screen with date, start/end time, and 2D/3D indicator; enforce capacity via seat inventory.
- ❖ Manage “currently playing” titles per theatre with run start/end and optional featured status.

4. Reservations and seat locking:

- ❖ Allow users to begin checkout by temporarily locking selected seats; locks carry start/expiry times and auto-release to prevent double-booking.
- ❖ Capture per-seat pricing and status within a booking; booking tracks timestamp, status, and total.

5. Bookings and unified payments:

- ❖ Support a single payment per booking that itemizes ticket subtotal, snack subtotal, tax, and total for auditability.
- ❖ Enable mixed tenders: credit cards (with card type, postal, CVV, expiry), gift cards with balance transactions, and Scene+ point redemptions tied to the payment.

6. Loyalty and memberships:

- ❖ Manage CineClub memberships with type, start/renewal and active status; issue member tickets with special pricing/terms.
- ❖ Maintain Scene+ accounts with point balances and redemption history connected to payments for integrity.

7. Concessions and orders:

- ❖ Store a snack catalog (items and combos) with availability and prices; record snack orders, items, quantities, and subtotals.
- ❖ Process snack orders at the theatre level and present tickets and snacks together in a consolidated order summary for user history.

8. Data integrity and rules:

- ❖ Enforce screen capacity through seat-level inventory; expire locks; validate membership active status; maintain gift card and points balances; and apply pricing via seat category modifiers.
- ❖ Ensure all entities are linked to theatre/showtime context for traceability across locations.

9. Out-of-scope items:

- ❖ Exclude staff/HR modules, third-party gateway or loyalty API integrations, real-time concession stock management, analytics dashboards, and frontend/mobile UI; focus on data model, relational design, sample data, and SQL deliverables per the final report template.

Functional Requirements

1. Movie & Content Discovery:

Browse available movies, view details (title, synopsis, rating, cast, genres, runtime, language), and search by genre or showtime.

2. Theatre & Showtime Management:

Display theatres with locations and screens; schedule showtimes per screen with 2D/3D flags and available seat counts; manage currently playing titles with run dates and featured designation.

3. Seat Selection & Reservation:

Display seat maps by availability status; lock selected seats temporarily during checkout to prevent double-booking; apply seat-category pricing (regular, VIP, D-Box, accessible).

2. Booking & Checkout:

Create bookings with multiple seats; calculate total price (ticket subtotal + seat modifiers); track booking status from creation to confirmation.

4. Multi-Method Payment Processing:

Accept credit card, gift card, and Scene+ point redemptions in a single transaction; itemize ticket and snack subtotals with tax; generate payment records and receipts.

5. Loyalty & Membership:

Enroll users in Scene+ rewards program and CineClub membership; track point balances and redemption history; issue member-exclusive tickets and pricing.

6. Concessions Ordering:

Browse snack catalog (items, combos, prices); add items to order; calculate snack subtotal, process snack orders per theatre location.

7. User Account Management:

Register and maintain user profiles (personal info, contact details, payment methods); store gift card and loyalty card associations; track order history.

8. Reporting & Admin Functions:

Generate seat occupancy reports by showtime; summarize ticket and snack sales; track payment method usage and loyalty program engagement.

Database Modelling

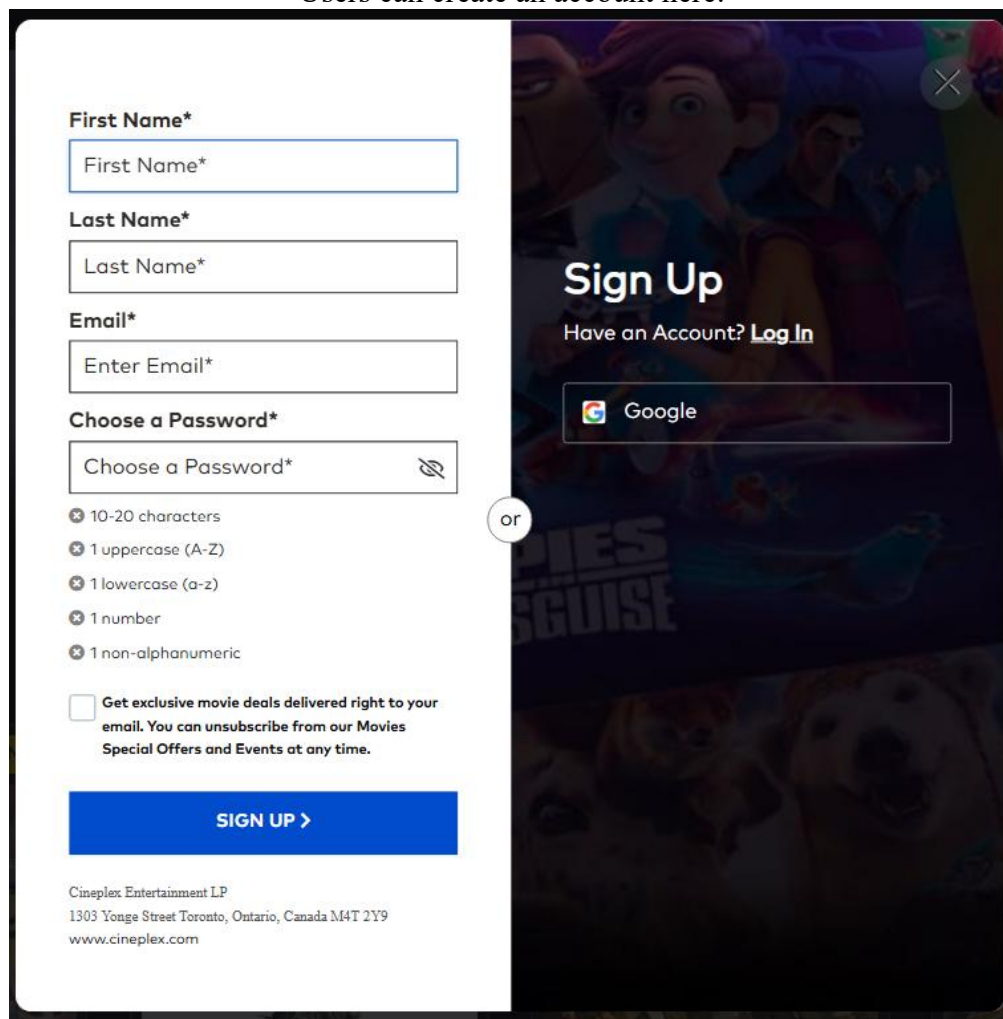
External Views

This is a step-by-step **Moviegoer** journey, showcasing the use of database.

1. Account Creation / Registration

- Navigate to registration/login screen.
- Input name, contact info, and password.
- Two-factor authentication (2FA) has been excluded for simplification purposes.
- Confirm registration and access user dashboard.

Users can create an account here:



The image shows a registration form for Cineplex. The form is on the left, and a background image of a movie poster is on the right. The form fields are:

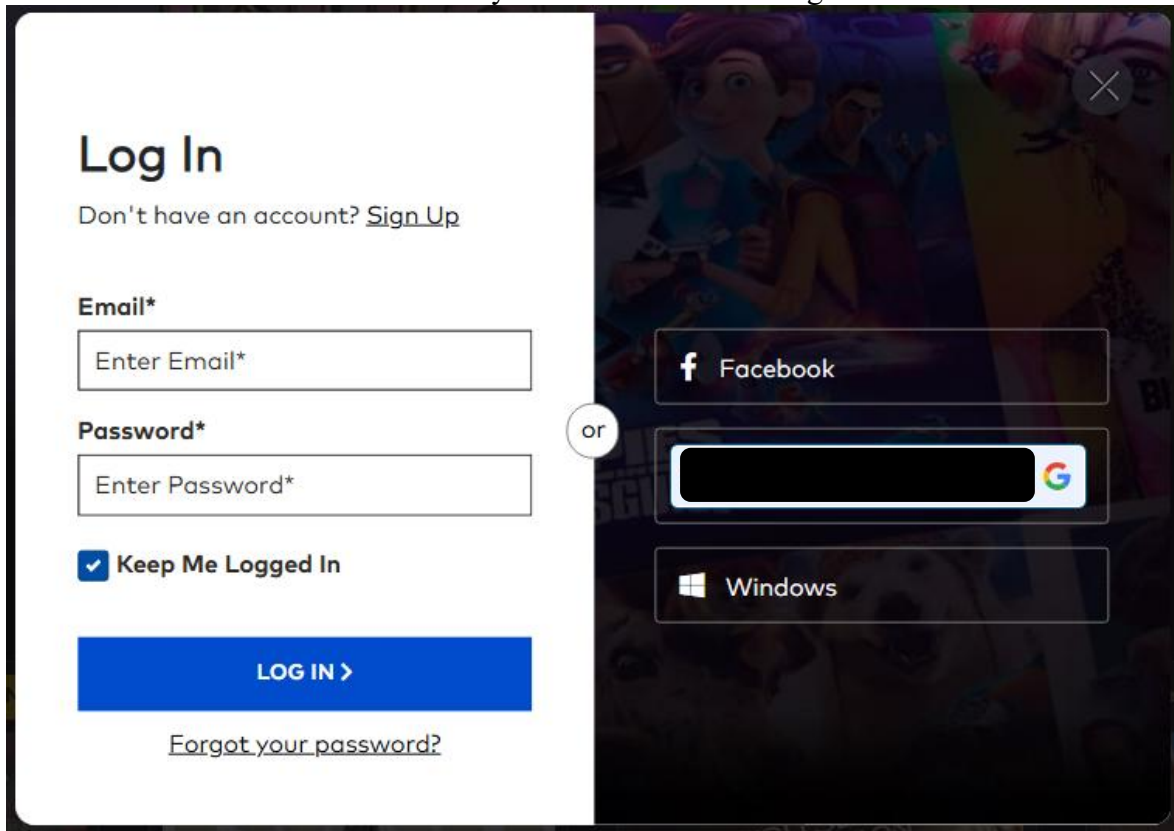
- First Name***: Text input field.
- Last Name***: Text input field.
- Email***: Text input field.
- Choose a Password***: Text input field with a password strength indicator icon.
- Password Requirements**:
 - 10-20 characters
 - 1 uppercase (A-Z)
 - 1 lowercase (a-z)
 - 1 number
 - 1 non-alphanumeric
- ☐ **Get exclusive movie deals delivered right to your email. You can unsubscribe from our Movies Special Offers and Events at any time.**
- SIGN UP >**: Blue button.

Below the form, the following text is displayed:

Cineplex Entertainment LP
1303 Yonge Street Toronto, Ontario, Canada M4T 2Y9
www.cineplex.com

The background image on the right shows a movie poster with the title "PIES IN THE GUISE" and a "Sign Up" overlay. The overlay includes the text "Have an Account? [Log In](#)" and a "Google" login button.

Users who already have an account can login here:



Log In

Don't have an account? [Sign Up](#)

Email*


Password*


☒ **Keep Me Logged In**


LOG IN >

[Forgot your password?](#)

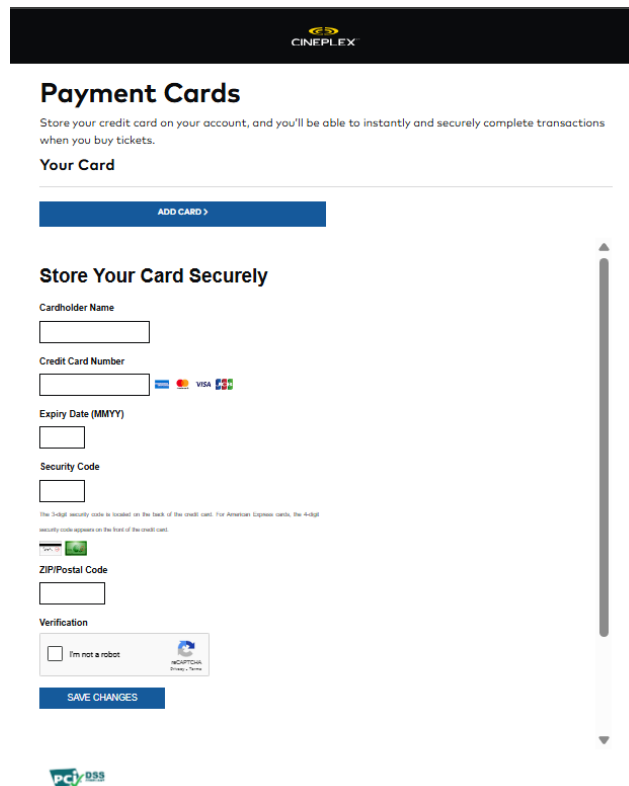
or

 **Facebook**



 **Windows**

Link Credit Cards/Gift Cards (Optional)



CINEPLEX

Payment Cards

Store your credit card on your account, and you'll be able to instantly and securely complete transactions when you buy tickets.





Your Card

ADD CARD >

Store Your Card Securely

Cardholder Name

Credit Card Number


Expiry Date (MMYY)

Security Code


The 3-digit security code is located on the back of the credit card. For American Express cards, the 4-digit security code appears on the front of the credit card.

ZIP/Postal Code

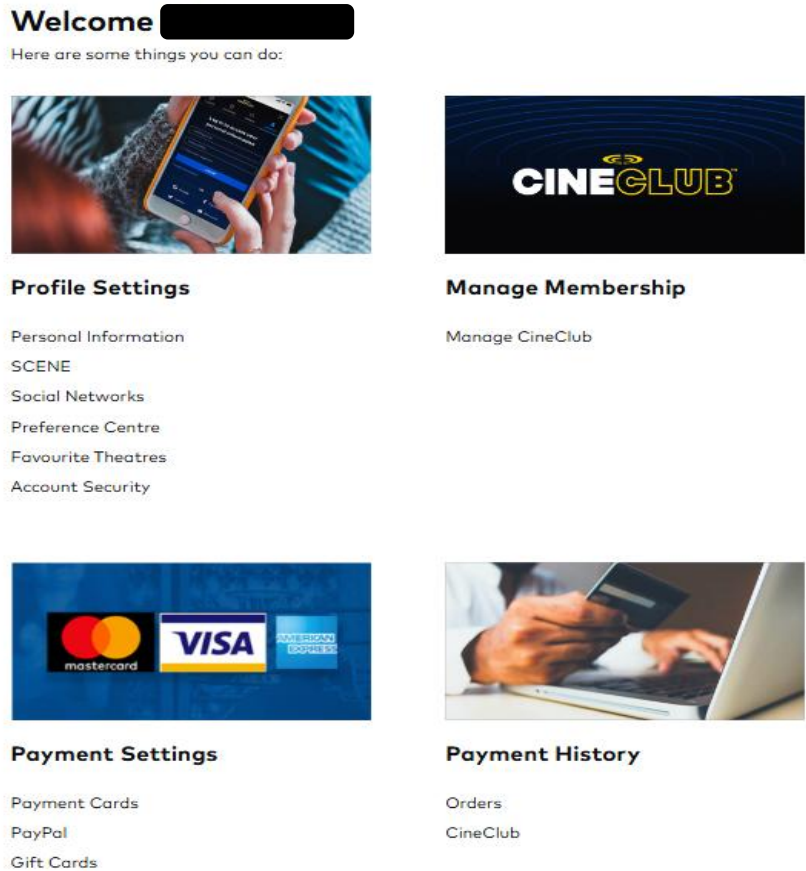
Verification

☐ I'm not a robot 

SAVE CHANGES



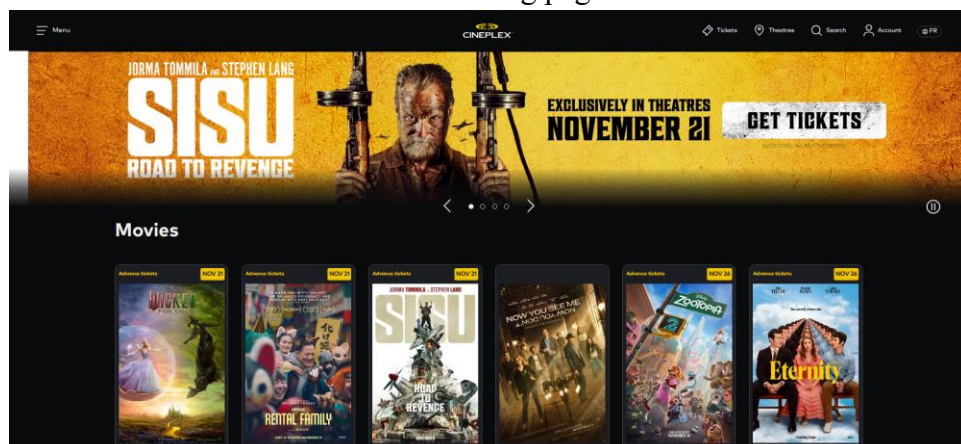
After Registration Users can access the account dashboard



2. Browse Cinemas and Movies

- Select theatre location or search by city.
- View list of currently playing movies.
- Click on a movie for details (synopsis, cast, genre, rating).
- Browse trailers and posters.

User landing page



Select theatre location or city

Select a theatre

Search by theatres or cities

[See All Theatres Nearby](#)

☆ **Cineplex Cinemas Vaughan**
4km Vaughan

☆ **Cineplex Cinemas Yonge-Eglinton and VIP**
12km Toronto

☆ **Cineplex VIP Cinemas Don Mills (age restricted 19+)**
14km Toronto

☆ **Cineplex Cinemas Yonge-Dundas and VIP**
16km Toronto

☆ **Cineplex Cinemas Yorkdale**
7km Toronto

☆ **SilverCity Richmond Hill Cinemas**
12km Richmond Hill

☆ **Cineplex Cinemas Varsity and VIP**
15km Toronto

☆ **Cineplex Cinemas Empress Walk**
9km Toronto

☆ **Cineplex Cinemas Fairview Mall**
14km Toronto

☆ **Cineplex Cinemas Queensway and VIP**
15km Toronto

View a list of currently playing movies

Cineplex Cinemas Vaughan


☆ [ADD AS FAVOURITE](#)

3555 Highway 7 West, Vaughan, ON, L4L 6B1 (905) 851-1001


ULTRA **AVX** | **IMAX** | **SCREEN X** | **REAL D 3D** | **D-BOX** | **ATMOS**

[Get Tickets](#)


Community Day Films




Sing 2
November 15th @ 10 am



Puss in Boots: The Last Wish
November 15th @ 10:20 am





The Wild Robot
November 15th @ 10:40 am





Minions: The Rise of Gru
November 15th @ 11:00 am


Movies






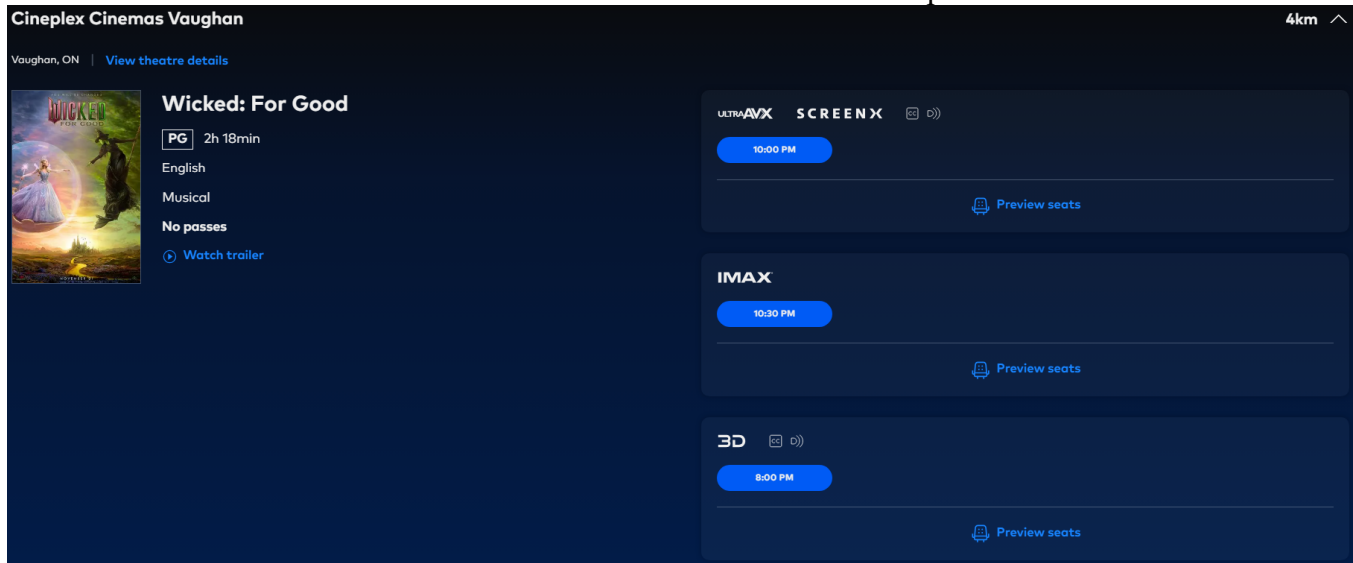








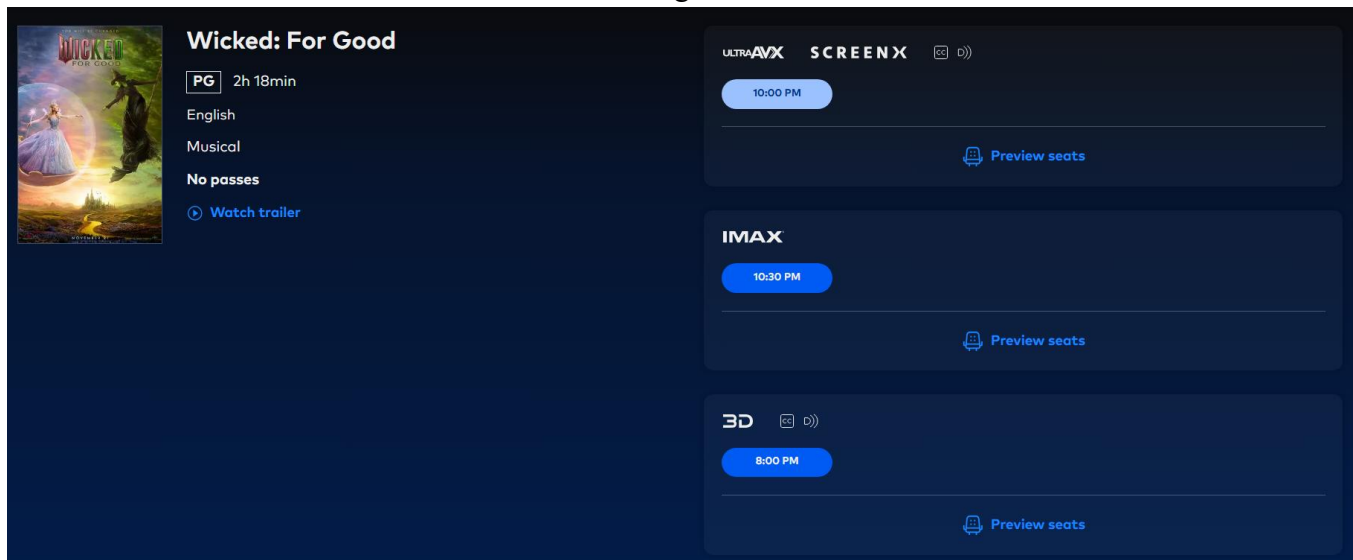
Select a movie for details and browse trailers/posters



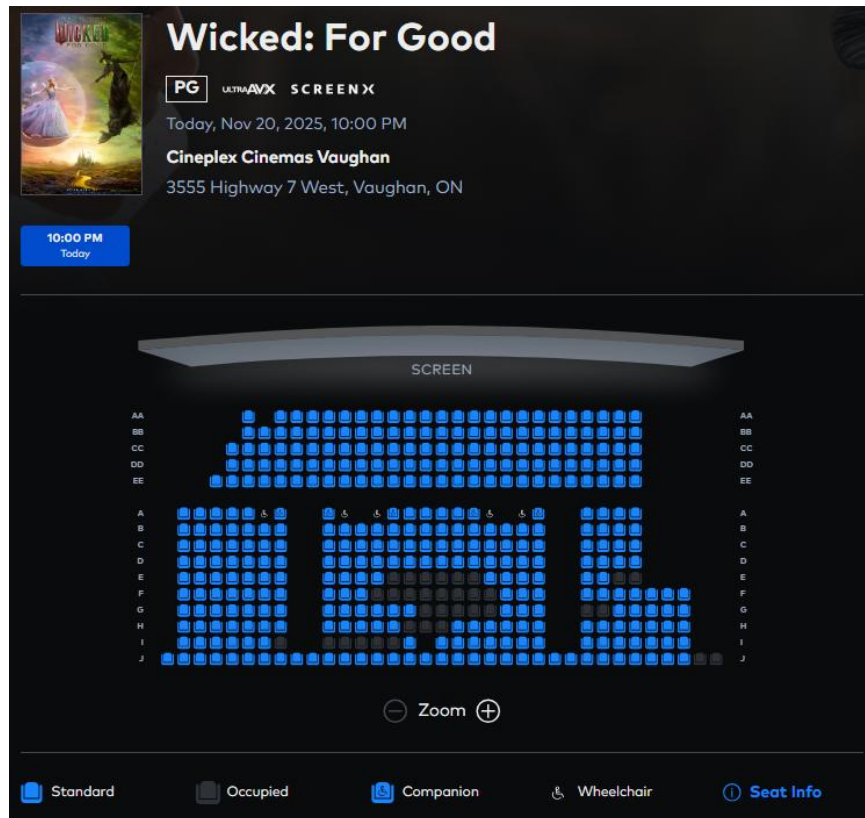
3. Choose Screening and View Seats

- Select a showtime and hall for the chosen movie.
- View the seat map, displaying occupied/reserved seats.
- Seat categories and badges are visible.

Selecting a show



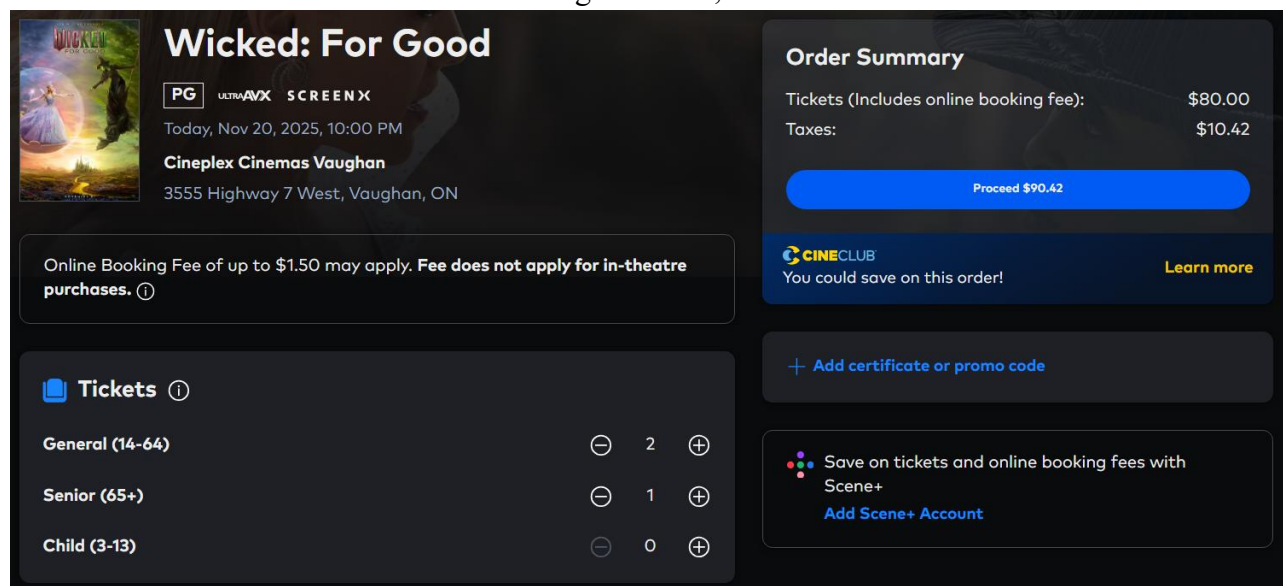
View the seat map, along with categories/badges




4. Select Seats and Tickets

- Select the desired number and type of seats.
- The seat lock is placed.
- View the seat map and select desired seat location.

After selecting the show, select tickets



View the seat map and reservations



Wicked: For Good


PG **ULTRA**AVX **SCREEN**X

Today, Nov 20, 2025, 10:00 PM

Cineplex Cinemas Vaughan
3555 Highway 7 West, Vaughan, ON

Seat Selection

4:40



Order Summary


Tickets (Includes online booking fee): \$80.00
Taxes: \$10.42

[Checkout \\$90.42](#)

Seats

G20 G21 G22

The seat lock prevents double reservation



Wicked: For Good


PG **ULTRA**AVX **3D** **D-BOX** **Dolby** Atmos

Tomorrow, Nov 21, 2025, 10:00 PM

Cineplex Cinemas Yonge-Eglinton and VIP
2300 Yonge Street, Toronto, ON

Seat Selection

4:07



Order Summary

Tickets (Includes online booking fee): \$0.00
Taxes: \$0.00

[Proceed \\$0.00](#)

D-BOX Tickets

D-BOX AVX General (14-64) 0
D-BOX AVX Senior (65+) 0
D-BOX AVX Child (3-13) 0

Standard Tickets

General (14-64) 0
Senior (65+) 0
Child (3-13) 0

Order Summary

Tickets (Includes online booking fee): \$0.00
Taxes: \$0.00

[Proceed \\$0.00](#)


CINECLUB

You could save on this order! [Learn more](#)

+ [Add certificate or promo code](#)

[Save on tickets and online booking fees with Scene+](#)
[Add Scene+ Account](#)

D-Box-enabled movies include D-Box ticket selections alongside standard options



Wicked: For Good


PG **ULTRA**AVX **3D** **D-BOX** **Dolby** Atmos

Tomorrow, Nov 21, 2025, 10:00 PM

Cineplex Cinemas Yonge-Eglinton and VIP
2300 Yonge Street, Toronto, ON

Seat Selection

4:07



Order Summary

Tickets (Includes online booking fee): \$0.00
Taxes: \$0.00

[Proceed \\$0.00](#)

Online Booking Fee of up to \$1.50 may apply. Fee does not apply for in-theatre purchases. ⓘ

D-BOX Tickets

D-BOX AVX General (14-64) 0
D-BOX AVX Senior (65+) 0
D-BOX AVX Child (3-13) 0

Standard Tickets

General (14-64) 0
Senior (65+) 0
Child (3-13) 0

Order Summary

Tickets (Includes online booking fee): \$0.00
Taxes: \$0.00

[Proceed \\$0.00](#)

CINECLUB

You could save on this order! [Learn more](#)

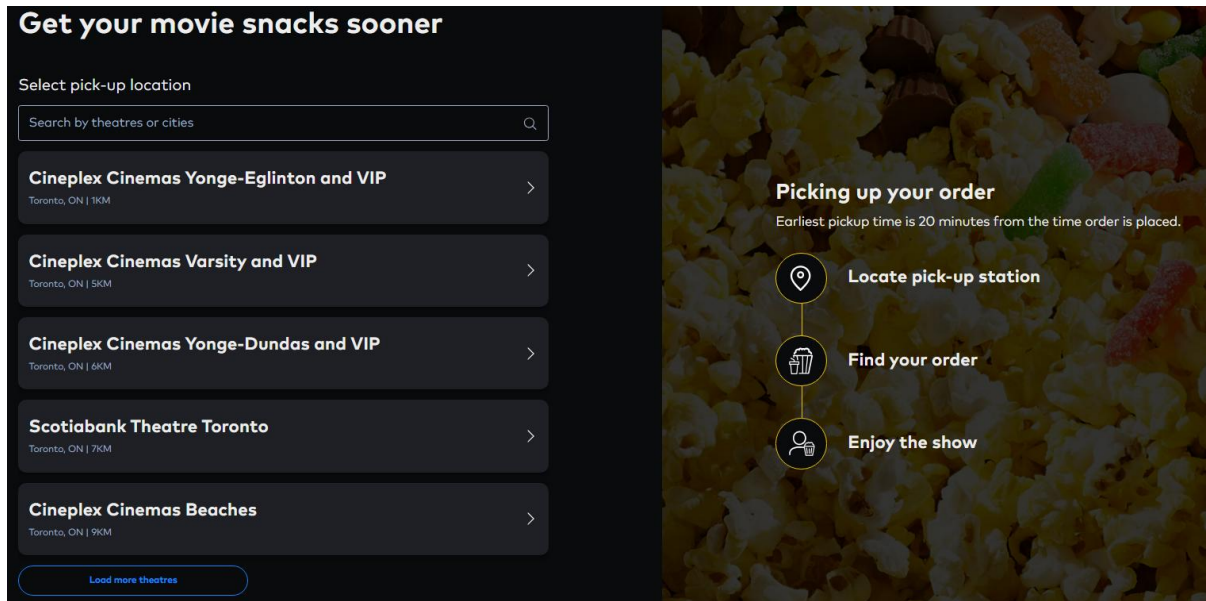
+ [Add certificate or promo code](#)

[Save on tickets and online booking fees with Scene+](#)
[Add Scene+ Account](#)

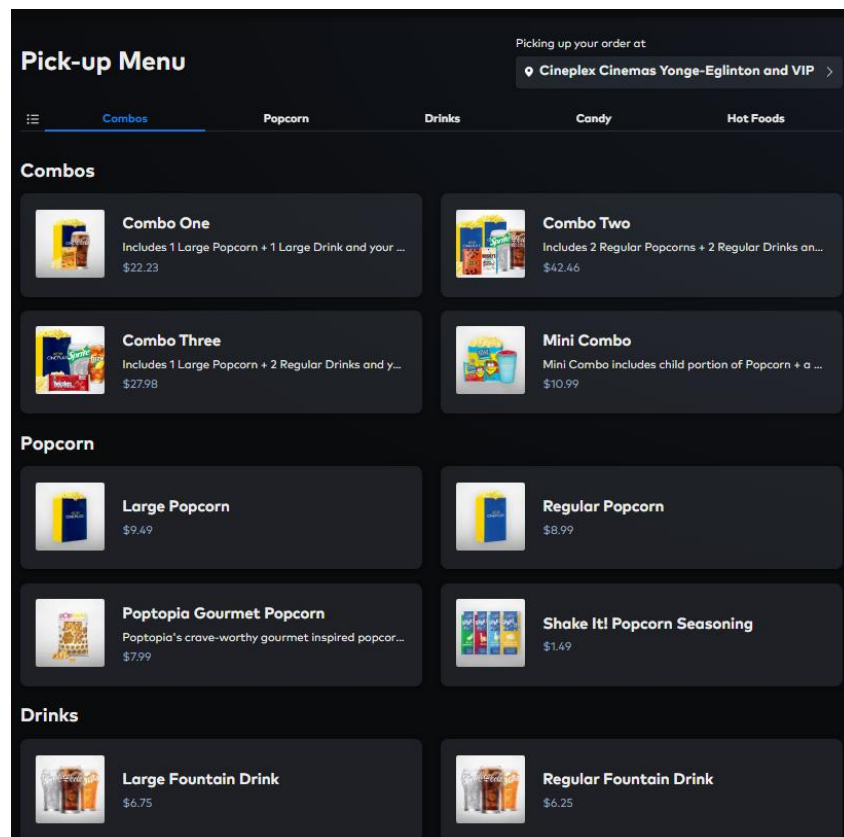
5. Add Snacks (Optional)

- Cineplex users can also browse and order snacks tied to the selected theatre
- Items are structured into categories and priced per unit
- Orders can include multiple items with custom quantities


Select a theatre



A wide selection of combos & snacks to choose from



Add snacks to your cart, which will be stored in the database



200 BONUS SCENE+ PTS

Combo One

Includes 1 Large Popcorn + 1 Large Drink and your choice of 1 Select Candy.

\$22.23 · 1400 Cals - 2080 Cals

− 2 +

Add for \$50.42

Large Popcorn 880 Cals


Butter on Top	+\$1.49	−	1	+
190 Cals				
Layered Butter	+\$2.98	−	0	+
380 Cals				
Shake It! Seasoning - Dill Pickle	+\$1.49	−	1	+
70 Cals				
Shake It! Seasoning - Ketchup	+\$1.49	−	0	+
70 Cals				
Shake It! Seasoning - Salt n' Vinegar	+\$1.49	−	0	+
70 Cals				
Shake It! Seasoning -				

6. Apply Offers and Pay

- Review cart: tickets, snacks, subtotal, tax, and total.
- Apply CineClub ticket, ScenePlus points, or gift card.
- Enter or select payment method (credit card, etc.).
- Submit payment and receive confirmation.

Review the snack cart

Your Cart 1 ITEM



Combo One
\$22.23

− 1 +

LARGE POPCORN

DRINKS

Coke Large x1 (\$0.00)

CANDY

Reese Mini x1 (\$0.00)

Edit Remove

Add more items

Subtotal: \$22.23

Taxes: \$2.89

Total: \$25.12

Apply gift cards/CineClub or select payment method

Checkout

Pick-up details

Cineplex Cinemas Yonge-Eglinton and VIP

Today, Nov 20 at 8:40 PM Change

Select other payment at next step >

Add gift card **NEW** >

Review the tickets cart/Add gift cards

WICKED

PG

ULTRA AVX 3D Dolby Atmos

Tomorrow, Nov 21, 2025, 10:00 PM

Cineplex Cinemas Yonge-Eglinton and VIP
2300 Yonge Street, Toronto, ON

Add gift card

Checkout
4:50

Order Summary

Tickets:

\$68.00

Online Booking Fee: ①

\$4.50

Taxes:

\$9.44

Proceed to pay \$81.94

By continuing past this page, you agree to our [Terms & Conditions](#)

Your Tickets (3)

Edit

1 x DBOX AVX General (14-64)

1 x General (14-64) (AVX)

1 x Senior (65+) (AVX)

Seat Numbers: F16, F17, I16

Proceed to payment with Credit Card

CINEPLEX

Total Amount: CAD 81.94

Pay with your Credit Card

Cardholder Name

Credit Card Number

Expiry Date (MMYY)

Security Code

The 3-digit security code is located on the back of the credit card. For American Express cards, the 4-digit security code appears on the front of the credit card.

ZIP/Postal Code

Pay With Your Credit Card

Pay with PayPal

PayPal

Your private information is secured using TLS 1.2 (Transport Layer Security). All information is encrypted between your web browser and E-xact Transactions (Canada) Ltd.

16

7. Confirmation and History

- Once payment succeeds, an OrderSummary is generated.
- This receipt captures the full breakdown of tickets, snack items, and applied discounts.
- Users can view their past orders in their account.

User can see previous reservations and order summary



Order History

View your Cineplex order history. Access your transaction history for Tickets, Digital Movies, Gift Cards and Food & Beverage.

Filter your Order History

Filter By Order Type

MOVIE TICKETS

Filter By Year

2025 & COMING

Purchased on: Jul 7, 2025, 03:06 PM
Booking ID: WLCK88Z



Superman - An IMAX 3D Experience®
Cineplex Cinemas Vaughan
Jul 15, 2025 | 4:10 PM
IMAX

1 X CineClub Member ,seat(s):J-14
1 X Free General Admit ,seat(s):J-13
1 X Free General Admit ,seat(s):J-12

Purchased on: May 3, 2025, 08:11 PM
Booking ID: WLRS378



Thunderbolts - An IMAX 3D Experience®
Cineplex Cinemas Vaughan
May 6, 2025 | 3:30 PM
IMAX

1 X SCENE+ TUE M ,seat(s):J-19
RATED:

Purchased on: Jun 30, 2025, 06:51 PM
Booking ID: WPDJNTF



How to Train Your Dragon
Cineplex Cinemas Yorkdale
Jul 1, 2025 | 6:20 PM
Aud 07

1 X Free General Admit ,seat(s):J-16
1 X Free General Admit ,seat(s):J-15
1 X SCENE+ TUE M ,seat(s):J-14

Purchased on: Apr 28, 2025, 01:48 PM
Booking ID: WJ6GD6W



Sinners
Scotiabank Theatre Toronto
Apr 29, 2025 | 4:30 PM
Aud 14

1 X SCENE+ TUE ,seat(s):J-14
1 X SCENE+ TUE ,seat(s):J-13
1 X SCENE+ TUE ,seat(s):J-12
RATED:

Purchased on: Jun 5, 2025, 04:16 PM
Booking ID: WTQGX6H



Mission: Impossible The Final Reckoning
Cineplex Cinemas Yorkdale
Jun 6, 2025 | 6:00 PM
AVX #10

Purchased on: Apr 13, 2025, 02:45 PM
Booking ID: WH7K5NX



A Minecraft Movie
Cineplex Cinemas Vaughan
May 12, 2025 | 3:30 PM

Other External Views as such:

1) Staff / Box Office

a) Find Showtimes and Seats

- Search for screening by movie, hall, or date.
- View real-time seat availability and locks.

b) Book/Edit on Behalf of Customer

- Select seats and create new booking.
- Modify or cancel existing bookings.

c) Accept Multiple Payments

- Process payments by credit card, gift card, or loyalty points.
- Issue and print booking receipt.

2) Concessions Staff

a) View Snack Inventory

- Access snack list and combo options by location.
- View prices and current availability.

b) Fulfill Snack Orders

- Review pending snack orders with quantities.
- Mark items as fulfilled and confirm payment.

3) Manager / Corporate

a) Screening Oversight

- Monitor active movies and showtime schedules by location.
- Review seat occupancy reports for each hall.

b) Review Sales & Loyalty Engagement

- Generate reports for ticket and snack sales totals.
- Track loyalty membership and redemption activity.

Key Functionalities of the System

Functions	Description
Movie Ticket Booking	Book tickets for any showtime in any theatre
Snack Ordering	Order food/drink items tied to theatre location
Real-Time Seat Selection	See live seat maps; temporary locks to prevent conflicts
Order Summary Generation	Unified view of completed bookings and orders
Secure Payment Processing	Support for credit card, gift card, and SCENE+ redemptions
Loyalty Programs	CineClub and SCENE+ programs fully integrated
User Account Management	One account per email; must register to access system
Movie Catalog	Browse films with posters, trailers, cast, and genres

Business Rules & Assumptions

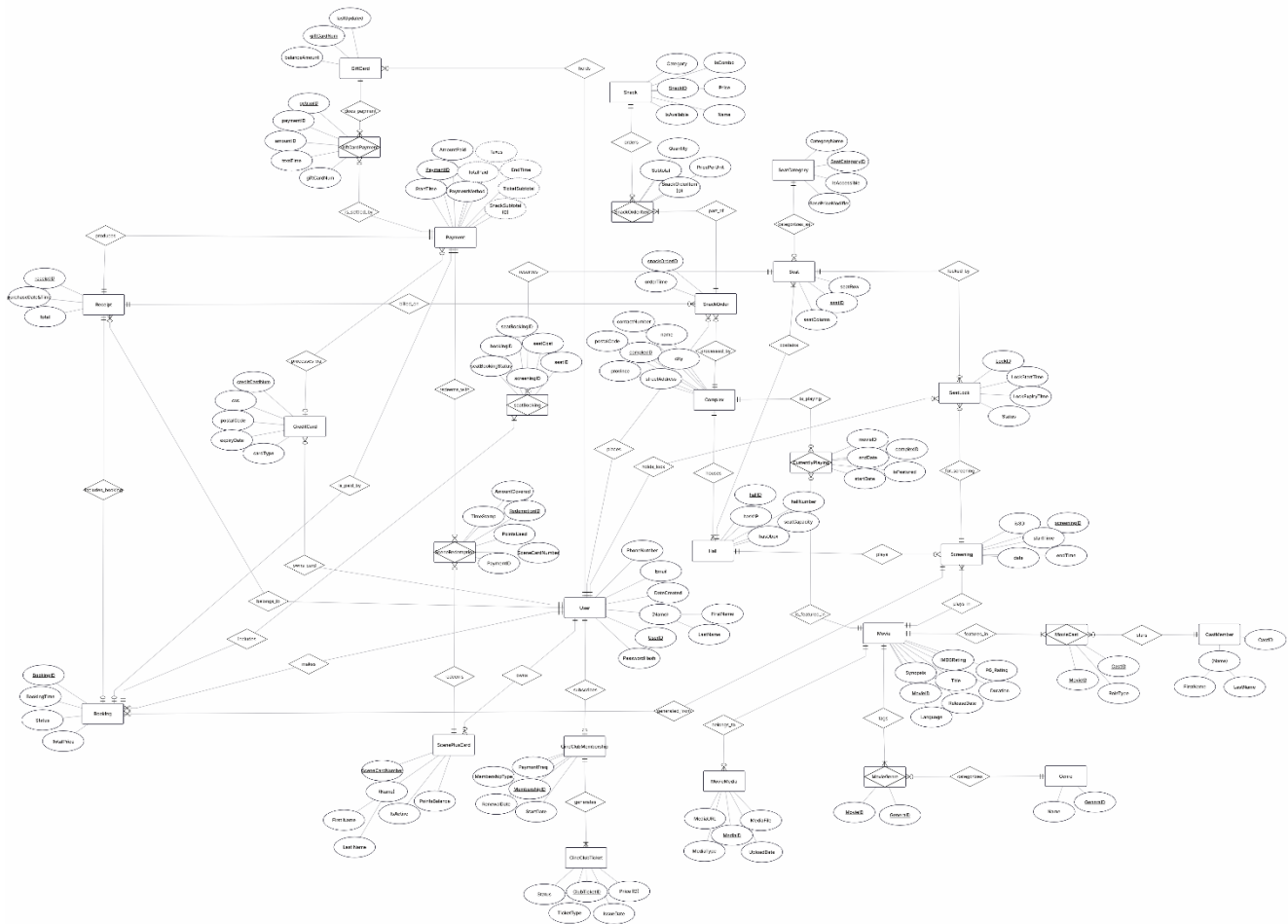
Business Rules

- User accounts are mandatory for all transactions, no guest or anonymous checkout.
- Seat locking prevents double-booking, locks expire automatically after 10 minutes, freeing seats for other users.
- Seat pricing is dynamic, calculated as base Screening price plus SeatCategory modifiers.
- One active booking per user per screening, users cannot book the same screening twice simultaneously.
- Payments support multiple concurrent methods, credit card, gift card balance deduction, and Scene Plus point redemption, within a single transaction.
- Each user can hold at most one active CineClub Membership and many ScenePlus Card at any time.
- All transactions generate immutable receipts, no post-payment refunds, cancellations, or transaction reversals are supported.
- Snack orders are fulfilled at the Complex level where they were placed.
- Movies can belong to multiple genres, cast members can appear in multiple movies.
- Currently playing status is managed per Complex, with featured designation and run date tracking.

Assumptions

1. Seat capacity is enforced at the Hall level with a fixed, unchangeable layout.
2. Snack availability is binary (available/unavailable), real-time inventory depletion is not tracked.
3. Payment records are created separately if a user purchases both tickets and snacks in one checkout.
4. Group bookings are not supported. Each Booking belongs to exactly one User.
5. Card data is encrypted/tokenized in production, prototype stores plaintext for demonstration.
6. Derived attributes (lock expiry, renewal dates, tax, total paid) are calculated automatically by the system.

The E-R Data Model



[Click here to access the High-Definition E-R model.](#)

Logical design of the Relational Database



Database design and implementation

Table design and data integrity control:

1. User

(UserID, Name, Email, PhoneNumber, PasswordHash, DateCreated)

Primary Key: UserID

Unique: Email

Domain Constraints:

- Email must be valid format
- PhoneNumber optional
- PasswordHash non-null
- DateCreated auto-timestamp

Normalization: 3NF - All attributes are atomic, depend on PK only.

2. CreditCard

(CreditCardNum, CardType, ExpiryDate, CVC, PostalCode)

Primary Key: CreditCardNum

Foreign Key: None (implicit link via Payment)

Domain Constraints:

- CreditCardNum 16 digits (encrypted storage)
- CVC 3-4 digits
- ExpiryDate MM/YY format, must be future date
- CardType: Visa, Mastercard, Amex
- PostalCode alphanumeric

Normalization: 3NF - All attributes atomic, non-key attributes depend only on PK.

3. ScenePlusCard

(SceneCardNumber, PointsBalance, IsActive)

Primary Key: SceneCardNumber

Foreign Key: UserID (via User relationship)

Domain Constraints:

- SceneCardNumber unique, alphanumeric
- PointsBalance ≥ 0 , default 0
- IsActive boolean, default True

Normalization: 3NF - non-key attributes depend directly on SceneCardNumber.

4. GiftCard

(GiftCardNum, BalanceAmount, LastUpdated)

Primary Key: GiftCardNum

Domain Constraints:

- GiftCardNum unique, 16 digits
- BalanceAmount ≥ 0 (currency)
- LastUpdated auto-timestamp on each transaction

Normalization: 3NF - All attributes atomic and depend on PK.

5. CineClubMembership

(MembershipID, UserID, MembershipType, StartDate, RenewalDate, PaymentFrequency)

Primary Key: MembershipID

Foreign Key: UserID \rightarrow User(UserID)

Unique: UserID (one membership per user)

Domain Constraints:

- MembershipType: Standard, Premium, VIP
- StartDate \leq RenewalDate
- PaymentFrequency: Monthly, Annual
- RenewalDate derived/calculated from StartDate + frequency

Normalization: 3NF - non-key attributes depend only on MembershipID; no transitive dependencies.

6. CineClubTicket

(ClubTicketID, MembershipID, TicketType, Status, Price, IssueDate)

Primary Key: ClubTicketID

Foreign Key: MembershipID \rightarrow CineClubMembership(MembershipID)

Domain Constraints:

- TicketType: Complimentary, Discounted, Premium
- Status: Active, Redeemed, Expired
- Price ≥ 0
- IssueDate auto-timestamp

Normalization: 3NF - All non-key attributes depend on ClubTicketID only.

7. Complex

(ComplexID, Name, StreetAddress, City, Province, PostalCode, ContactNumber)

Primary Key: ComplexID

Domain Constraints:

- Name unique within city/province
- StreetAddress non-null
- PostalCode alphanumeric
- ContactNumber format validated

Normalization: 3NF - All attributes atomic; each describes a unique theatre location.

8. Hall

(HallID, ComplexID, HallNumber, SeatCapacity, HasVIP, HasDBox)

Primary Key: HallID

Foreign Key: ComplexID -> Complex(ComplexID)

Unique: (ComplexID, HallNumber)

Domain Constraints:

- SeatCapacity > 0
- HasVIP, HasDBox boolean

Normalization: 3NF - non-key attributes depend on HallID; ComplexID is FK (not transitive).

9. SeatCategory

(SeatCategoryID, CategoryName, BasePriceModifier, IsAccessible)

Primary Key: SeatCategoryID

Unique: CategoryName

Domain Constraints:

- BasePriceModifier decimal (e.g., 1.0 for standard, 1.5 for VIP)
- IsAccessible boolean

Normalization: 3NF - All attributes atomic and depend on SeatCategoryID only.

10. Seat

(SeatID, HallID, SeatRow, SeatColumn, SeatCategoryID)

Primary Key: SeatID

Foreign Keys:

- HallID -> Hall(HallID)

- SeatCategoryID -> SeatCategory(SeatCategoryID)

Unique: (HallID, SeatRow, SeatColumn)

Domain Constraints:

- SeatRow alphanumeric (A, B, C, ...)
- SeatColumn numeric (1, 2, 3, ...)

Normalization: 3NF - non-key attributes (SeatRow, SeatColumn, SeatCategoryID) depend on Seat only; FKs not transitive.

11. Movie

(MovieID, Title, Synopsis, IMDBRating, Language, ReleaseDate, Duration, PG_Rating)

Primary Key: MovieID

Domain Constraints:

- Title non-null
- IMDBRating 0.0-10.0
- Language ISO 639-1 code (en, fr, etc.)
- Duration in minutes, > 0
- PG_Rating: G, PG, 12A, 15, 18, etc.

Normalization: 3NF - All attributes atomic, depend on MovieID only.

12. Genre

((GenreID, GenreName)

Primary Key: GenreID

Unique: GenreName

Domain Constraints:

- GenreName non-null

Normalization: 3NF - All attributes atomic and directly dependent on GenreID.

13. MovieGenre (Associative)

(MovieID, GenreID)

Primary Key: (MovieID, GenreID)

Foreign Keys:

- MovieID -> Movie(MovieID)
- GenreID -> Genre(GenreID)

Domain Constraints: None (junction table)

Normalization: 3NF - Only FKs; represents many-to-many without redundancy.

14. CastMember

(CastID, Name)

Primary Key: CastID

Domain Constraints:

- Name non-null

Normalization: 3NF - All attributes atomic and depend on CastID

15. MovieCast (Associative)

(MovieID, CastID, RoleType)

Primary Key: (MovieID, CastID)

Foreign Keys:

- MovieID -> Movie(MovieID)
- CastID -> CastMember(CastID)

Domain Constraints:

- RoleType: Actor, Director, Producer, etc.

Normalization: 3NF - All attributes atomic; resolves many-to-many relationship.

16. MovieMedia

(MediaID, MovieID, MediaType, MediaURL, MediaFile, UploadDate)

Primary Key: MediaID

Foreign Key: MovieID -> Movie(MovieID)

Domain Constraints:

- MediaType: Trailer, Poster, Clip, etc.
- MediaURL or MediaFile non-null
- UploadDate auto-timestamp

Normalization: 3NF - non-key attributes depend on MediaID; MovieID is FK.

17. Screening

(ScreeningID, HallID, MovieID, ScreeningDate, StartTime, EndTime, Is3D)

Primary Key: ScreeningID

Foreign Keys:

- HallID -> Hall(HallID)
- MovieID -> Movie(MovieID)

Domain Constraints:

- ScreeningDate >= current date (for future screenings)

- $StartTime < EndTime$
- Is3D boolean

Normalization: 3NF - All non-key attributes depend on ScreeningID; FKs not transitive.

18. CurrentlyPlaying (Associative)

(ComplexID, MovieID, IsFeatured, StartDate, EndDate)

Primary Key: (ComplexID, MovieID)

Foreign Keys:

- ComplexID \rightarrow Complex(ComplexID)
- MovieID \rightarrow Movie(MovieID)

Domain Constraints:

- $StartDate \leq EndDate$
- IsFeatured boolean

Normalization: 3NF - Only PK and FKs + descriptive attributes; no transitive dependencies.

19. SeatLock

(LockID, UserID, SeatID, ScreeningID, LockStartTime, LockExpiryTime, Status)

Primary Key: LockID

Foreign Keys:

- UserID \rightarrow User(UserID)
- SeatID \rightarrow Seat(SeatID)
- ScreeningID \rightarrow Screening(ScreeningID)

Domain Constraints:

- $LockStartTime \leq LockExpiryTime$
- $LockExpiryTime = LockStartTime + 10 \text{ minutes}$ (derived)
- Status: Active, Expired, Released

Normalization: 3NF - All attributes atomic; no transitive dependencies.

20. Booking

(BookingID, UserID, ScreeningID, BookingTime, Status, TotalPrice)

Primary Key: BookingID

Foreign Keys:

- UserID \rightarrow User(UserID)
- ScreeningID \rightarrow Screening(ScreeningID)

Domain Constraints:

- BookingTime auto-timestamp
- Status: Pending, Confirmed, Cancelled, Expired
- TotalPrice ≥ 0 (currency)

Normalization: 3NF - non-key attributes depend on BookingID; FKs are non-transitive.

21. seatBooking (Associative)

(seatBookingID, BookingID, SeatID, ScreeningID, SeatCost, SeatBookingStatus)

Primary Key: seatBookingID

Foreign Keys:

- BookingID -> Booking(BookingID)
- SeatID -> Seat(SeatID)
- ScreeningID -> Screening(ScreeningID)

Domain Constraints:

- SeatCost ≥ 0 (calculated as Seat.SeatCategory.BasePriceModifier * Screening base price)
- SeatBookingStatus: Booked, Cancelled, Refunded

Normalization: 3NF - Resolves many-to-many; SeatCost computed at booking time and stored.

22. Payment

(PaymentID, BookingID, SnackOrderID, PaymentMethod, StartTime, EndTime, AmountPaid, Taxes, TicketSubtotal, SnackSubtotal, TotalPaid)

Primary Key: PaymentID

Foreign Keys:

- BookingID -> Booking(BookingID) [optional, nullable]
- SnackOrderID -> SnackOrder(SnackOrderID) [optional, nullable]

Domain Constraints:

- PaymentMethod: CreditCard, GiftCard, ScenePlusCard, Multiple
- StartTime \leq EndTime
- AmountPaid, Taxes, Subtotals ≥ 0
- TotalPaid = TicketSubtotal + SnackSubtotal + Taxes (derived)
- At least one of BookingID or SnackOrderID non-null

Normalization: 3NF - All non-key attributes depend on PaymentID; derived fields calculated.

23. CreditCardPayment (Associative/Implicit)

(PaymentID, CreditCardNum, AmountCharged)

Primary Key: PaymentID

Foreign Keys:

- PaymentID -> Payment(PaymentID)
- CreditCardNum -> CreditCard(CreditCardNum)

Domain Constraints:

- AmountCharged > 0

Normalization: 3NF - Tracks credit card used for a payment.

24. GiftCardPayment (Associative)

(GcTrxnID, PaymentID, GiftCardNum, AmountDeducted, TransactionTime)

Primary Key: GcTrxnID

Foreign Keys:

- PaymentID -> Payment(PaymentID)
- GiftCardNum -> GiftCard(GiftCardNum)

Domain Constraints:

- AmountDeducted > 0, <= GiftCard.BalanceAmount
- TransactionTime auto-timestamp

Normalization: 3NF - Each transaction logged separately; GiftCard balance updated after.

25. SceneRedemption (Associative)

(RedemptionID, PaymentID, SceneCardNumber, PointsUsed, AmountCovered, RedemptionTime)

Primary Key: RedemptionID

Foreign Keys:

- PaymentID -> Payment(PaymentID)
- SceneCardNumber -> ScenePlusCard(SceneCardNumber)

Domain Constraints:

- PointsUsed > 0
- AmountCovered > 0 (discount value applied)
- RedemptionTime auto-timestamp

Normalization: 3NF - Each redemption tracked; ScenePlusCard.PointsBalance updated after.

26. Snack

(SnackID, SnackName, Category, Price, IsCombo, IsAvailable)

Primary Key: SnackID

Domain Constraints:

- SnackName non-null
- Category: Beverage, Popcorn, Candy, Combo, etc.
- Price > 0 (currency)
- IsCombo boolean (indicates if bundle of items)
- IsAvailable boolean, default True

Normalization: 3NF - All attributes atomic and depend on SnackID only.

27. SnackOrder

(SnackOrderID, UserID, ComplexID, OrderTime)

Primary Key: SnackOrderID

Foreign Keys:

- UserID -> User(UserID)
- ComplexID -> Complex(ComplexID)

Domain Constraints:

- OrderTime auto-timestamp

Normalization: 3NF - non-key attributes depend on SnackOrderID; ComplexID specifies fulfillment location.

28. SnackOrderItem (Associative)

(SnackOrderItemID, SnackOrderID, SnackID, Quantity, PricePerUnit, Subtotal)

Primary Key: SnackOrderItemID

Foreign Keys:

- SnackOrderID -> SnackOrder(SnackOrderID)
- SnackID -> Snack(SnackID)

Domain Constraints:

- Quantity > 0 (integer)
- PricePerUnit > 0 (captured at order time for audit)
- Subtotal = Quantity * PricePerUnit (derived, stored for historical accuracy)

Normalization: 3NF - Resolves many-to-many; subtotal stored to preserve historical pricing.

29. Receipt

(ReceiptID, PaymentID, BookingID, SnackOrderID, PurchaseDateTime, Total)

Primary Key: ReceiptID

Foreign Keys:

- PaymentID -> Payment(PaymentID)
- BookingID -> Booking(BookingID) [optional]
- SnackOrderID -> SnackOrder(SnackOrderID) [optional]

Domain Constraints:

- PurchaseDateTime auto-timestamp (matches Payment.EndTime)
- Total = Payment.TotalPaid (derived/denormalized for convenience)
- Immutable record (no updates after creation)

Normalization: 3NF (slight denormalization for convenience) - Receipt consolidates Payment + associated orders for customer history.

Normalization Summary

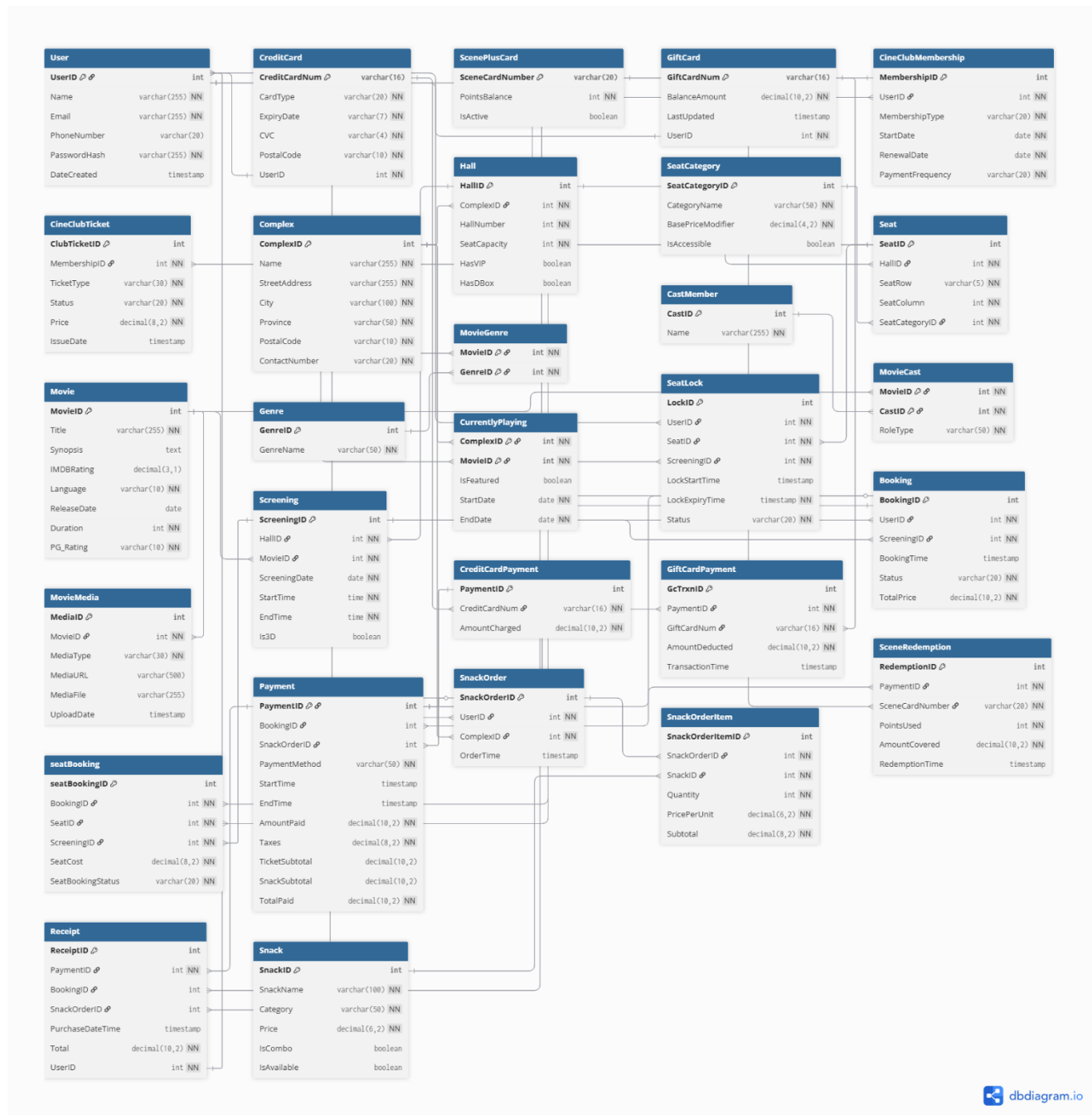
All 29 tables are normalized to **3NF (Third Normal Form)**:

1. **1NF**: All attributes are atomic; no repeating groups. Multi-valued relationships (e.g., Movie-Genre, Movie-Cast) are resolved via associative tables.
2. **2NF**: All non-key attributes depend fully on the primary key, not on partial keys. Composite keys in associative tables satisfy this requirement.
3. **3NF**: No transitive dependencies. Non-key attributes depend only on the primary key, not on other non-key attributes.

Denormalizations (Intentional for Performance/Audit):

- **PricePerUnit & Subtotal** in SnackOrderItem: Stored to preserve historical pricing despite being derivable.
- **Total** in Receipt: Denormalized from Payment.TotalPaid for convenience in receipt generation.
- **SeatCost** in seatBooking: Stored at booking time to maintain historical audit trail.
- **Derived fields** (TotalPaid, LockExpiryTime, RenewalDate, Tax): Calculated on-the-fly or stored as needed for reporting/auditing.

The following design relational diagram presents a clearer and more detailed design and implementation of our database:



(All diagrams of this report are available on the drive in high-resolution; [Click here!](#))

Following are the Table Creation queries for all 29 tables in the database:

(The SQL file can be found in the drive!)

```
CREATE TABLE [User] (  
    UserID INT PRIMARY KEY IDENTITY(1,1),  
    [Name] NVARCHAR(255) NOT NULL,  
    Email NVARCHAR(255) UNIQUE NOT NULL,  
    PhoneNumber NVARCHAR(20),  
    PasswordHash NVARCHAR(255) NOT NULL,  
    DateCreated DATETIME2 DEFAULT GETDATE()  
);  
CREATE INDEX idx_email ON [User](Email);  
CREATE INDEX idx_name ON [User]([Name]);
```

```
CREATE TABLE CreditCard (  
    CreditCardNum NVARCHAR(16) PRIMARY KEY,  
    UserID INT NOT NULL,  
    CardType NVARCHAR(20) NOT NULL,  
    ExpiryDate NVARCHAR(7) NOT NULL,  
    CVC NVARCHAR(4) NOT NULL,  
    PostalCode NVARCHAR(10) NOT NULL,  
    CONSTRAINT FK_CreditCard_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE NO  
ACTION ON UPDATE NO ACTION  
);
```

```
CREATE TABLE ScenePlusCard (  
    SceneCardNumber NVARCHAR(20) PRIMARY KEY,  
    PointsBalance INT NOT NULL DEFAULT 0 CHECK (PointsBalance >= 0),  
    IsActive BIT NOT NULL DEFAULT 1  
);
```

```
CREATE TABLE GiftCard (  
    GiftCardNum NVARCHAR(16) PRIMARY KEY,  
    UserID INT NOT NULL,  
    BalanceAmount DECIMAL(10, 2) NOT NULL CHECK (BalanceAmount >= 0),  
    LastUpdated DATETIME2 DEFAULT GETDATE(),  
    CONSTRAINT FK_GiftCard_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE NO ACTION  
ON UPDATE NO ACTION  
);
```

```
CREATE TABLE CineClubMembership (  
    MembershipID INT PRIMARY KEY IDENTITY(1,1),  
    UserID INT UNIQUE NOT NULL,  
    MembershipType NVARCHAR(20) NOT NULL,  
    StartDate DATE NOT NULL,  
    RenewalDate DATE NOT NULL,  
    PaymentFrequency NVARCHAR(20) NOT NULL,  
    CONSTRAINT FK_CineClubMembership_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

```

CREATE TABLE CineClubTicket (
    ClubTicketID INT PRIMARY KEY IDENTITY(1,1),
    MembershipID INT NOT NULL,
    TicketType NVARCHAR(30) NOT NULL,
    [Status] NVARCHAR(20) NOT NULL,
    Price DECIMAL(8, 2) NOT NULL,
    IssueDate DATETIME2 DEFAULT GETDATE(),
    CONSTRAINT FK_CineClubTicket_Membership FOREIGN KEY (MembershipID) REFERENCES
CineClubMembership(MembershipID) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE [Complex] (
    ComplexID INT PRIMARY KEY IDENTITY(1,1),
    [Name] NVARCHAR(255) NOT NULL,
    StreetAddress NVARCHAR(255) NOT NULL,
    City NVARCHAR(100) NOT NULL,
    Province NVARCHAR(50) NOT NULL,
    PostalCode NVARCHAR(10) NOT NULL,
    ContactNumber NVARCHAR(20) NOT NULL
);
CREATE INDEX idx_location ON [Complex](City, Province);

```

```

CREATE TABLE Hall (
    HallID INT PRIMARY KEY IDENTITY(1,1),
    ComplexID INT NOT NULL,
    HallNumber INT NOT NULL,
    SeatCapacity INT NOT NULL CHECK (SeatCapacity > 0),
    HasVIP BIT DEFAULT 0,
    HasDBox BIT DEFAULT 0,
    CONSTRAINT FK_Hall_Complex FOREIGN KEY (ComplexID) REFERENCES [Complex](ComplexID) ON DELETE
CASCADE ON UPDATE CASCADE,
    CONSTRAINT UK_Hall_Number UNIQUE (ComplexID, HallNumber)
);

```

```

CREATE TABLE SeatCategory (
    SeatCategoryID INT PRIMARY KEY IDENTITY(1,1),
    CategoryName NVARCHAR(50) UNIQUE NOT NULL,
    BasePriceModifier DECIMAL(4, 2) NOT NULL,
    IsAccessible BIT DEFAULT 0
);

```

```

CREATE TABLE Seat (
    SeatID INT PRIMARY KEY IDENTITY(1,1),
    HallID INT NOT NULL,
    SeatRow NVARCHAR(5) NOT NULL,
    SeatColumn INT NOT NULL,
    SeatCategoryID INT NOT NULL,
    CONSTRAINT FK_Seat_Hall FOREIGN KEY (HallID) REFERENCES Hall(HallID) ON DELETE CASCADE ON
UPDATE CASCADE,
    CONSTRAINT FK_Seat_Category FOREIGN KEY (SeatCategoryID) REFERENCES SeatCategory(SeatCategoryID) ON
DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT UK_Seat_Location UNIQUE (HallID, SeatRow, SeatColumn)
);

CREATE TABLE Movie (
    MovieID INT PRIMARY KEY IDENTITY(1,1),
    Title NVARCHAR(255) NOT NULL,
    Synopsis NVARCHAR(MAX),
    IMDBRating DECIMAL(3, 1),
    [Language] NVARCHAR(10) NOT NULL,
    ReleaseDate DATE,
    Duration INT NOT NULL CHECK (Duration > 0),
    PG_Rating NVARCHAR(10) NOT NULL
);

CREATE TABLE [Genre] (
    GenreID INT PRIMARY KEY IDENTITY(1,1),
    GenreName NVARCHAR(50) UNIQUE NOT NULL
);

CREATE TABLE MovieGenre (
    MovieID INT NOT NULL,
    GenreID INT NOT NULL,
    PRIMARY KEY (MovieID, GenreID),
    CONSTRAINT FK_MovieGenre_Movie FOREIGN KEY (MovieID) REFERENCES Movie(MovieID) ON DELETE
CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_MovieGenre_Genre FOREIGN KEY (GenreID) REFERENCES [Genre](GenreID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE CastMember (
    CastID INT PRIMARY KEY IDENTITY(1,1),
    [Name] NVARCHAR(255) NOT NULL
);

```

```

CREATE TABLE MovieCast (
    MovieID INT NOT NULL,
    CastID INT NOT NULL,
    RoleType NVARCHAR(50) NOT NULL,
    PRIMARY KEY (MovieID, CastID),
    CONSTRAINT FK_MovieCast_Movie FOREIGN KEY (MovieID) REFERENCES Movie(MovieID) ON DELETE
    CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_MovieCast_Cast FOREIGN KEY (CastID) REFERENCES CastMember(CastID) ON DELETE
    CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE MovieMedia (
    MediaID INT PRIMARY KEY IDENTITY(1,1),
    MovieID INT NOT NULL,
    MediaType NVARCHAR(30) NOT NULL,
    MediaURL NVARCHAR(500),
    MediaFile NVARCHAR(255),
    UploadDate DATETIME2 DEFAULT GETDATE(),
    CONSTRAINT FK_MovieMedia_Movie FOREIGN KEY (MovieID) REFERENCES Movie(MovieID) ON DELETE
    CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE Screening (
    ScreeningID INT PRIMARY KEY IDENTITY(1,1),
    HallID INT NOT NULL,
    MovieID INT NOT NULL,
    ScreeningDate DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    Is3D BIT DEFAULT 0,
    CONSTRAINT FK_Screening_Hall FOREIGN KEY (HallID) REFERENCES Hall(HallID) ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT FK_Screening_Movie FOREIGN KEY (MovieID) REFERENCES Movie(MovieID) ON DELETE NO
    ACTION ON UPDATE NO ACTION,
    CONSTRAINT CHK_Screening_Times CHECK (StartTime < EndTime)
);
CREATE INDEX idx_screening_schedule ON Screening(HallID, ScreeningDate, StartTime);

```

```

CREATE TABLE CurrentlyPlaying (
    ComplexID INT NOT NULL,
    MovieID INT NOT NULL,
    IsFeatured BIT DEFAULT 0,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    PRIMARY KEY (ComplexID, MovieID),
    CONSTRAINT FK_CurrentlyPlaying_Complex FOREIGN KEY (ComplexID) REFERENCES [Complex](ComplexID)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_CurrentlyPlaying_Movie FOREIGN KEY (MovieID) REFERENCES Movie(MovieID) ON DELETE
    CASCADE ON UPDATE CASCADE,
    CONSTRAINT CHK_Playing_Dates CHECK (EndDate >= StartDate)
);

```

```

);

CREATE TABLE SeatLock (
    LockID INT PRIMARY KEY IDENTITY(1,1),
    UserID INT NOT NULL,
    SeatID INT NOT NULL,
    ScreeningID INT NOT NULL,
    LockStartTime DATETIME2 DEFAULT GETDATE(),
    LockExpiryTime DATETIME2 NOT NULL,
    [Status] NVARCHAR(20) NOT NULL,
    CONSTRAINT FK_SeatLock_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT FK_SeatLock_Seat FOREIGN KEY (SeatID) REFERENCES Seat(SeatID) ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT FK_SeatLock_Screening FOREIGN KEY (ScreeningID) REFERENCES Screening(ScreeningID) ON
    DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT CHK_Lock_Expiry CHECK (LockExpiryTime > LockStartTime)
);
CREATE INDEX idx_seat_screening ON SeatLock(SeatID, ScreeningID);

CREATE TABLE Booking (
    BookingID INT PRIMARY KEY IDENTITY(1,1),
    UserID INT NOT NULL,
    ScreeningID INT NOT NULL,
    BookingTime DATETIME2 DEFAULT GETDATE(),
    [Status] NVARCHAR(20) NOT NULL,
    TotalPrice DECIMAL(10, 2) NOT NULL CHECK (TotalPrice >= 0),
    CONSTRAINT FK_Booking_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT FK_Booking_Screening FOREIGN KEY (ScreeningID) REFERENCES Screening(ScreeningID) ON
    DELETE NO ACTION ON UPDATE NO ACTION
);
CREATE INDEX idx_user_booking ON Booking(UserID, ScreeningID);

CREATE TABLE seatBooking (
    seatBookingID INT PRIMARY KEY IDENTITY(1,1),
    BookingID INT NOT NULL,
    SeatID INT NOT NULL,
    ScreeningID INT NOT NULL,
    SeatCost DECIMAL(8, 2) NOT NULL CHECK (SeatCost >= 0),
    SeatBookingStatus NVARCHAR(20) NOT NULL,
    CONSTRAINT FK_SeatBooking_Booking FOREIGN KEY (BookingID) REFERENCES Booking(BookingID) ON
    DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_SeatBooking_Seat FOREIGN KEY (SeatID) REFERENCES Seat(SeatID) ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT FK_SeatBooking_Screening FOREIGN KEY (ScreeningID) REFERENCES Screening(ScreeningID) ON
    DELETE NO ACTION ON UPDATE NO ACTION
);
CREATE INDEX idx_booking_seat ON seatBooking(BookingID, SeatID);

```

```

CREATE TABLE Snack (
    SnackID INT PRIMARY KEY IDENTITY(1,1),
    SnackName NVARCHAR(100) NOT NULL,
    Category NVARCHAR(50) NOT NULL,
    Price DECIMAL(6, 2) NOT NULL CHECK (Price > 0),
    IsCombo BIT DEFAULT 0,
    IsAvailable BIT DEFAULT 1
);

CREATE TABLE SnackOrder (
    SnackOrderID INT PRIMARY KEY IDENTITY(1,1),
    UserID INT NOT NULL,
    ComplexID INT NOT NULL,
    OrderTime DATETIME2 DEFAULT GETDATE(),
    CONSTRAINT FK_SnackOrder_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_SnackOrder_Complex FOREIGN KEY (ComplexID) REFERENCES [Complex](ComplexID) ON DELETE NO ACTION ON UPDATE NO ACTION
);

CREATE TABLE SnackOrderItem (
    SnackOrderItemID INT PRIMARY KEY IDENTITY(1,1),
    SnackOrderID INT NOT NULL,
    SnackID INT NOT NULL,
    Quantity INT NOT NULL CHECK (Quantity > 0),
    PricePerUnit DECIMAL(6, 2) NOT NULL CHECK (PricePerUnit > 0),
    Subtotal DECIMAL(8, 2) NOT NULL CHECK (Subtotal > 0),
    CONSTRAINT FK_SnackOrderItem_Order FOREIGN KEY (SnackOrderID) REFERENCES SnackOrder(SnackOrderID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_SnackOrderItem_Snack FOREIGN KEY (SnackID) REFERENCES Snack(SnackID) ON DELETE NO ACTION ON UPDATE NO ACTION
);

CREATE TABLE [Payment] (
    PaymentID INT PRIMARY KEY IDENTITY(1,1),
    BookingID INT,
    SnackOrderID INT,
    PaymentMethod NVARCHAR(50) NOT NULL,
    StartTime DATETIME2 DEFAULT GETDATE(),
    EndTime DATETIME2,
    AmountPaid DECIMAL(10, 2) NOT NULL CHECK (AmountPaid >= 0),
    Taxes DECIMAL(8, 2) NOT NULL CHECK (Taxes >= 0),
    TicketSubtotal DECIMAL(10, 2),
    SnackSubtotal DECIMAL(10, 2),
    TotalPaid DECIMAL(10, 2) NOT NULL CHECK (TotalPaid >= 0),
    CONSTRAINT CHK_Payment_Link CHECK (BookingID IS NOT NULL OR SnackOrderID IS NOT NULL),
    CONSTRAINT CHK_Total_Paid CHECK (TotalPaid = ISNULL(TicketSubtotal, 0) + ISNULL(SnackSubtotal, 0) + Taxes),

```

```

    CONSTRAINT FK_Payment_Booking FOREIGN KEY (BookingID) REFERENCES Booking(BookingID) ON DELETE
    NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_Payment_SnackOrder FOREIGN KEY (SnackOrderID) REFERENCES SnackOrder(SnackOrderID)
    ON DELETE NO ACTION ON UPDATE NO ACTION
);
CREATE INDEX idx_payment_booking ON [Payment](BookingID);
CREATE INDEX idx_payment_snack ON [Payment](SnackOrderID);

```

```

CREATE TABLE CreditCardPayment (
    PaymentID INT PRIMARY KEY,
    CreditCardNum NVARCHAR(16) NOT NULL,
    AmountCharged DECIMAL(10, 2) NOT NULL CHECK (AmountCharged > 0),
    CONSTRAINT FK_CreditCardPayment_Payment FOREIGN KEY (PaymentID) REFERENCES [Payment](PaymentID)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_CreditCardPayment_Card FOREIGN KEY (CreditCardNum) REFERENCES
    CreditCard(CreditCardNum) ON DELETE NO ACTION ON UPDATE NO ACTION
);

```

```

CREATE TABLE GiftCardPayment (
    GcTrxnID INT PRIMARY KEY IDENTITY(1,1),
    PaymentID INT NOT NULL,
    GiftCardNum NVARCHAR(16) NOT NULL,
    AmountDeducted DECIMAL(10, 2) NOT NULL CHECK (AmountDeducted > 0),
    TransactionTime DATETIME2 DEFAULT GETDATE(),
    CONSTRAINT FK_GiftCardPayment_Payment FOREIGN KEY (PaymentID) REFERENCES [Payment](PaymentID) ON
    DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_GiftCardPayment_Card FOREIGN KEY (GiftCardNum) REFERENCES GiftCard(GiftCardNum) ON
    DELETE NO ACTION ON UPDATE NO ACTION
);

```

```

CREATE TABLE SceneRedemption (
    RedemptionID INT PRIMARY KEY IDENTITY(1,1),
    PaymentID INT NOT NULL,
    SceneCardNumber NVARCHAR(20) NOT NULL,
    PointsUsed INT NOT NULL CHECK (PointsUsed > 0),
    AmountCovered DECIMAL(10, 2) NOT NULL CHECK (AmountCovered > 0),
    RedemptionTime DATETIME2 DEFAULT GETDATE(),
    CONSTRAINT FK_SceneRedemption_Payment FOREIGN KEY (PaymentID) REFERENCES [Payment](PaymentID) ON
    DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_SceneRedemption_Card FOREIGN KEY (SceneCardNumber) REFERENCES
    ScenePlusCard(SceneCardNumber) ON DELETE NO ACTION ON UPDATE NO ACTION
);

```

```

CREATE TABLE Receipt (
    ReceiptID INT PRIMARY KEY IDENTITY(1,1),
    PaymentID INT UNIQUE NOT NULL,
    UserID INT NOT NULL,
    BookingID INT,
    SnackOrderID INT,
    PurchaseDateTime DATETIME2 DEFAULT GETDATE(),

```

```

    Total DECIMAL(10, 2) NOT NULL CHECK (Total >= 0),
    CONSTRAINT FK_Receipt_Payment FOREIGN KEY (PaymentID) REFERENCES [Payment](PaymentID) ON DELETE
    NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_Receipt_User FOREIGN KEY (UserID) REFERENCES [User](UserID) ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT FK_Receipt_Booking FOREIGN KEY (BookingID) REFERENCES Booking(BookingID) ON DELETE
    NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_Receipt_SnackOrder FOREIGN KEY (SnackOrderID) REFERENCES SnackOrder(SnackOrderID)
    ON DELETE NO ACTION ON UPDATE NO ACTION
);

```

```

CREATE INDEX idx_user_email ON [User](Email);
CREATE INDEX idx_booking_user_date ON Booking(UserID, BookingTime);
CREATE INDEX idx_seat_booking_status ON seatBooking(SeatBookingStatus);
CREATE INDEX idx_seat_lock_expiry ON SeatLock(LockExpiryTime);
CREATE INDEX idx_movie_title ON Movie(Title);
CREATE INDEX idx_movie_genre ON MovieGenre(GenreID);
CREATE INDEX idx_screening_movie ON Screening(MovieID);
CREATE INDEX idx_payment_method ON [Payment](PaymentMethod);
CREATE INDEX idx_payment_time ON [Payment](StartTime);
CREATE INDEX idx_snack_order_user ON SnackOrder(UserID);
CREATE INDEX idx_snack_order_complex ON SnackOrder(ComplexID);

```

Entity Integrity

Entity integrity ensures that each table in the database maintains a unique identifier for every record, and that no component of the primary key can be null. In our database design, we ensured entity integrity by rigorously assigning primary keys to every table and, where appropriate, using composite primary keys to model M:N relationships or context-specific uniqueness.

Examples of Composite Primary Keys in Our Design:

Table Name	Composite Key Field
BookingSeat	bookingID, seatID
SceneRedemption	SceneCardNumber, redemptionID
GiftCardPayment	GiftCardNum, paymentID
SnackOrderItem	snackOrderID, snackID

By applying these keys, we ensure that each record remains uniquely identifiable and prevents insertion of null values in primary key columns. For example, in the SnackOrderItem table, each snack item is uniquely tied to a specific order, avoiding any accidental duplication.

Domain Integrity

Domain integrity restricts attribute values to valid, meaningful, and defined formats. In our database, we enforced domain integrity by specifying:

- Appropriate data types (INT, VARCHAR, DECIMAL, DATE, etc.)
- NOT NULL constraints to ensure mandatory data
- CHECK constraints to enforce logical validity
- UNIQUE constraints for distinct attributes (like email)

Attribute	Constraint(s)	Description
email (User)	VARCHAR(100), NOT NULL	Valid format required
paymentMethod (Payment)	ENUM or VARCHAR, NOT NULL	Restricts values to defined payment modes
basePriceModifier	DECIMAL(4,2), CHECK (value > 0)	Prevents illogical seat pricing modifiers
quantity (SnackOrderItem)	INT, CHECK (quantity >= 1)	Avoids invalid snack order quantities

Referential Integrity

Referential integrity guarantees consistency in relationships between tables. It requires that every foreign key either matches an existing primary key in the referenced table or is set to NULL when the relationship is optional.

In our design, we applied foreign keys throughout and used actions like ON DELETE CASCADE or SET NULL, depending on the business logic, to ensure data integrity.

Examples:

Foreign Key (FK)	References (PK)	ON DELETE Action	Description
userID in Booking	User(userID)	CASCADE	Bookings tied to valid users
paymentID in Receipt	Payment(paymentID)	CASCADE	Ensures summary always tied to payment
seatID in seatBooking	Seat(seatID)	CASCADE	Validates assigned seats exist
snackOrderID in SnackOrderItem	SnackOrder(snackOrderID)	CASCADE	Validates items tied to existing orders
sceneCardNumber in SceneRedemption	ScenePlusCard(sceneCardNum)	SET NULL	Optional linkage to loyalty program

We ensured that optional relationships such as GiftCardPayment to Payment, or SceneRedemption to a Scene card, allow NULLs to reflect real-world optionality.

Sample Data for All the Tables

Genre Table:

GenreID	GenreName
1	Action
9	Adventure
7	Animation
2	Comedy
10	Documentary
3	Drama
4	Horror
5	Romance
6	Science Fiction
8	Thriller

SeatCategory Table:

SeatCategoryID	CategoryName	BasePrice	IsAccessible
1	Standard	1.00	0
2	VIP	1.50	0
3	D-Box Motion	1.75	0
4	Accessible	1.00	1

CastMember Table:

CastID	Name
1	Tom Cruise
2	Scarlett Johansson
3	Ryan Gosling
4	Emma Stone
5	Leonardo DiCaprio
6	Zendaya
7	Timothée Chalamet
8	Cillian Murphy
9	Robert Downey Jr.
10	Gal Gadot

Snack Table:

SnackID	SnackName	Category	Price	IsCombo	IsAvaila...
1	Small Popcorn	Popcorn	6.99	0	1
2	Large Popcorn	Popcorn	9.99	0	1
3	Buttered Popcorn	Popcorn	8.49	0	1
4	Small Coca-Cola	Beverage	5.99	0	1
5	Large Coca-Cola	Beverage	7.99	0	1
6	Gummy Bears	Candy	5.49	0	1
7	Chocolate Chip Cookie	Candy	4.99	0	1
8	Nachos with Cheese	Food	8.99	0	1
9	Hot Dog	Food	7.49	0	1
10	Combo - Popcorn + Drink	Combo	12.99	1	1
11	Ice Cream	Candy	6.99	0	1
12	Candy Mix	Candy	4.49	0	1
13	Water Bottle	Beverage	3.99	0	1
14	Nachos Cheese Sauce	Food	2.99	0	1
15	Monster Energy Drink	Beverage	6.99	0	1

User Table:

UserID	Name	Email	PhoneN...	PasswordHash	DateCreated
1	John Smith	john.smith@email.com	416-555-0100	hashed_pwd_123_secure	2025-11-21 07:11:51.7466667
2	Sarah Johnson	sarah.johnson@email.com	416-555-0101	hashed_pwd_456_secure	2025-11-21 07:11:51.7466667
3	Michael Chen	michael.chen@email.com	416-555-0102	hashed_pwd_789_secure	2025-11-21 07:11:51.7466667
4	Emily Davis	emily.davis@email.com	416-555-0103	hashed_pwd_012_secure	2025-11-21 07:11:51.7466667
5	Alex Williams	alex.williams@email.com	416-555-0104	hashed_pwd_345_secure	2025-11-21 07:11:51.7466667
6	Jessica Brown	jessica.brown@email.com	416-555-0105	hashed_pwd_678_secure	2025-11-21 07:11:51.7466667
7	David Martinez	david.martinez@email.com	416-555-0106	hashed_pwd_901_secure	2025-11-21 07:11:51.7466667

CreditCard Table:

CreditCardN...	UserID	CardType	ExpiryD...	CVC	PostalC...
378282246310005	3	Amex	06/27	1234	M1A 1A1
4532015112830366	1	Visa	12/26	123	M5H 2N2
4916338506082832	4	Visa	03/28	789	L5B 1C3
5105105105105100	5	Mastercard	09/26	321	M5V 1Y7
5425233010103442	2	Mastercard	08/25	456	M4B 1B3

GiftCard Table:

GiftCardNum	UserID	Balance...	LastUpdated
9000001234567890	1	150.00	2025-11-21 07:13:50.0300000
9000001234567891	2	75.50	2025-11-21 07:13:50.0300000
9000001234567892	3	200.00	2025-11-21 07:13:50.0300000
9000001234567893	4	50.00	2025-11-21 07:13:50.0300000
9000001234567894	5	100.00	2025-11-21 07:13:50.0300000
9000001234567895	6	300.00	2025-11-21 07:13:50.0300000

Movie Table;

MovieID	Title	Synopsis	IMDBRa...	Language	Release...	Duration	PG_Rati...
1	Inception	A skilled thief who steals corporate secrets through dream-sha...	8.8	en	2010-07-16	148	12A
2	The Shawshank Redemption	Two imprisoned men bond over a number of years, finding sol...	9.3	en	1994-10-14	142	15
3	Interstellar	A team of explorers travel through a wormhole in space in an ...	8.6	en	2014-11-07	169	12A
4	La La Land	While navigating their careers in Los Angeles, a pianist and an ...	8.0	en	2016-12-09	128	12A
5	Oppenheimer	The story of American scientist J. Robert Oppenheimer and his...	8.4	en	2023-07-21	180	15
6	Spider-Man Across the Spider-Verse	Miles and his allies journey across the multiverse.	8.5	en	2023-06-02	140	12A
7	Dune Part Two	Paul leads the Fremen in a war against those who destroyed hi...	8.1	en	2024-02-29	166	12A
8	Toy Story 4	When a new toy called Forky joins Woody and the gang, a roa...	7.7	en	2019-06-21	100	G
9	The Dark Knight	Batman faces the Joker, a criminal mastermind who wants to p...	9.0	en	2008-07-18	152	12A
10	Avatar The Way of Water	Jake Sully leads the Navi against a new threat on Pandora.	7.9	en	2022-12-16	192	12A

MovieGenre Table:

MovieID	GenreID
1	6
1	8
2	3
3	6
3	9
4	3
4	5
5	3
5	8
6	1
6	6
7	1
7	6
8	7
9	1
9	8
10	6
10	9

MovieCast Table:

MovieID	CastID	RoleType
1	1	Actor
3	5	Actor
4	3	Actor
4	4	Actor
5	8	Actor
6	6	Actor
7	7	Actor
9	5	Actor
10	2	Actor

MovieMedia Table: *(mediaFile column empty because we don't have movies downloaded)*

MediaID	MovieID	MediaT...	MediaURL	MediaFile	UploadDate
1	1	Poster	https://cineplex.com/media/inception-poster.jpg	NULL	2025-11-21 07:18:22.4466667
2	1	Trailer	https://cineplex.com/media/inception-trailer.mp4	NULL	2025-11-21 07:18:22.4466667
3	3	Poster	https://cineplex.com/media/interstellar-poster.jpg	NULL	2025-11-21 07:18:22.4466667
4	4	Poster	https://cineplex.com/media/lalaland-poster.jpg	NULL	2025-11-21 07:18:22.4466667
5	5	Poster	https://cineplex.com/media/oppenheimer-poster.jpg	NULL	2025-11-21 07:18:22.4466667
6	6	Poster	https://cineplex.com/media/spiderman-poster.jpg	NULL	2025-11-21 07:18:22.4466667
7	7	Poster	https://cineplex.com/media/dune-poster.jpg	NULL	2025-11-21 07:18:22.4466667
8	8	Poster	https://cineplex.com/media/toystory-poster.jpg	NULL	2025-11-21 07:18:22.4466667
9	9	Trailer	https://cineplex.com/media/darkknight-trailer.mp4	NULL	2025-11-21 07:18:22.4466667
10	10	Poster	https://cineplex.com/media/avatar-poster.jpg	NULL	2025-11-21 07:18:22.4466667

ScenePlusCard Table:

SceneC...	PointsB...	IsActive
SCENE001	500	1
SCENE002	250	1
SCENE003	1000	1
SCENE004	0	1
SCENE005	750	1
SCENE006	350	1
SCENE007	2000	1

CineClubMembership Table:

Membe...	UserID	Membe...	StartDate	Renewal...	Paymen...
1	1	Standard	2024-01-15	2025-01-15	Annual
2	2	Premium	2024-06-01	2025-06-01	Annual
3	4	VIP	2023-11-20	2024-11-20	Annual
4	5	Standard	2024-09-10	2025-09-10	Annual
5	6	Premium	2024-04-05	2025-04-05	Annual

CineClubTicket Table:

ClubTic...	Membe...	TicketTy...	Status	Price	IssueDate
1	1	Complimentary	Active	0.00	2025-11-21 07:21:50.0766667
2	2	Discounted	Active	8.99	2025-11-21 07:21:50.0766667
3	3	Premium	Active	14.99	2025-11-21 07:21:50.0766667
4	1	Discounted	Redeemed	8.99	2025-10-22 07:21:50.0766667
5	4	Complimentary	Active	0.00	2025-11-21 07:21:50.0766667
6	5	Premium	Active	14.99	2025-11-21 07:21:50.0766667

Complex Table:

Comple...	Name	StreetAddress	City	Province	PostalC...	Contact...
1	Cineplex Odeon Yonge-Dundas	10 Dundas Street West	Toronto	ON	M5G 2C2	416-515-0500
2	Cineplex Odeon Scotiabank Theatre	259 Richmond Street West	Toronto	ON	M5V 1Y7	416-863-0500
3	Cineplex Odeon Forum Theatre	1 York Street	Toronto	ON	M5J 0N6	416-815-0500
4	Cineplex Odeon Mississauga	1 Dundas Street West	Mississauga	ON	L5B 1C3	905-271-1234

Hall Table:

HallID	Comple...	HallNu...	SeatCap...	HasVIP	HasDBox
1	1	1	200	1	0
2	1	2	180	0	1
3	1	3	150	1	1
4	2	1	250	1	1
5	2	2	180	0	0
6	3	1	120	1	0
7	4	1	200	0	1
8	4	2	150	1	0

Seat Table:

SeatID	HallID	SeatRow	SeatCol...	SeatCat...
1	1	A	1	1
2	1	A	2	1
3	1	A	3	1
4	1	A	4	1
5	1	A	5	1
6	1	B	1	1
7	1	B	2	2
8	1	B	3	2
9	1	B	4	2
10	1	B	5	1
11	1	C	1	1
12	1	C	2	1
13	1	C	3	3
14	1	C	4	3
15	1	C	5	1
16	1	D	1	4
17	1	D	2	1
18	1	D	3	1
19	1	D	4	1
20	1	D	5	4
21	2	A	1	1
22	2	A	2	1
23	2	A	3	1
24	2	A	4	1
25	2	A	5	1
26	2	B	1	2
27	2	B	2	2
28	2	B	3	2
29	2	B	4	2

Screening Table:

Screeni...	HallID	MovielD	Screeni...	StartTime	EndTime	Is3D
1	1	1	2025-11-21	14:00:00	15:48:00	0
2	1	1	2025-11-21	19:30:00	21:18:00	0
3	2	3	2025-11-21	15:00:00	16:49:00	1
4	2	3	2025-11-21	19:00:00	20:49:00	1
5	3	4	2025-11-21	16:30:00	18:38:00	0
6	4	5	2025-11-22	10:00:00	11:20:00	0
7	5	2	2025-11-22	14:00:00	15:42:00	0
8	1	7	2025-11-22	18:00:00	19:46:00	0
9	6	9	2025-11-22	16:00:00	17:32:00	0
10	7	10	2025-11-23	15:30:00	17:42:00	0

CurrentlyPlaying Table:

Comple...	MovieID	IsFeatur...	StartDate	EndDate
1	1	1	2025-11-21	2025-11-27
1	3	1	2025-11-21	2025-11-27
1	4	0	2025-11-21	2025-11-27
2	2	0	2025-11-22	2025-11-28
2	5	1	2025-11-22	2025-11-28
3	7	0	2025-11-21	2025-11-27
3	9	1	2025-11-22	2025-11-28
4	8	0	2025-11-21	2025-11-27

SeatLock Table:

LockID	UserID	SeatID	Screeni...	LockStartTime	LockExpiryTime	Status
1	1	1	1	2025-11-21 07:27:30.6500000	2025-11-21 07:37:30.6500000	Active
2	2	21	3	2025-11-21 07:12:30.6500000	2025-11-21 07:22:30.6500000	Expired
3	3	41	5	2025-11-21 07:22:30.6500000	2025-11-21 07:32:30.6500000	Active
4	5	2	1	2025-11-21 07:27:30.6500000	2025-11-21 07:37:30.6500000	Active
5	6	31	3	2025-11-21 07:07:30.6500000	2025-11-21 07:17:30.6500000	Expired

Booking Table:

Booking...	UserID	Screeni...	BookingTime	Status	TotalPrice
1	1	1	2025-11-21 07:28:29.5800000	Confirmed	29.98
2	2	3	2025-11-21 07:28:29.5800000	Confirmed	19.99
3	3	4	2025-11-21 07:28:29.5800000	Confirmed	39.97
4	4	5	2025-11-21 07:28:29.5800000	Pending	14.99
5	5	6	2025-11-21 07:28:29.5800000	Confirmed	44.96
6	6	9	2025-11-20 07:28:29.5800000	Confirmed	29.98
7	7	10	2025-11-19 07:28:29.5800000	Confirmed	34.97

seatBooking Table:

seatBoo...	Booking...	SeatID	Screeni...	SeatCost	SeatBoo...
1	1	1	1	14.99	Booked
2	1	2	1	14.99	Booked
3	2	21	3	19.99	Booked
4	3	26	4	19.99	Booked
5	3	27	4	19.99	Booked
6	4	41	5	14.99	Booked
7	5	6	6	22.48	Booked
8	5	7	6	22.48	Booked
9	6	51	9	14.99	Booked
10	7	31	10	17.49	Booked

SnackOrder Table:

SnackOr...	UserID	Comple...	OrderTime
1	1	1	2025-11-21 07:30:38.416667
2	2	1	2025-11-21 07:30:38.416667
3	3	2	2025-11-21 07:30:38.416667
4	5	1	2025-11-21 07:30:38.416667
5	6	3	2025-11-20 07:30:38.416667
6	7	4	2025-11-19 07:30:38.416667

SnackOrderItem Table:

SnackOr...	SnackOr...	SnackID	Quantity	PricePer...	Subtotal
1	1	1	2	6.99	13.98
2	1	4	2	5.99	11.98
3	2	10	1	12.99	12.99
4	3	3	1	8.49	8.49
5	3	5	1	7.99	7.99
6	4	1	1	6.99	6.99
7	4	4	1	5.99	5.99
8	5	2	1	9.99	9.99
9	5	6	2	5.49	10.98
10	6	10	1	12.99	12.99

Payment Table:

Paymen...	Booking...	SnackOr...	Paymen...	StartTime	EndTime	Amount...	Taxes	TicketSu...	SnackSu...	TotalPaid
1	1	NULL	CreditCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	29.98	3.90	29.98	NULL	33.88
2	2	NULL	ScenePlusCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	19.99	2.60	19.99	NULL	22.59
3	3	NULL	CreditCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	39.97	5.20	39.97	NULL	45.17
4	4	NULL	GiftCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	14.99	1.95	14.99	NULL	16.94
5	5	NULL	CreditCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	44.96	5.84	44.96	NULL	50.80
6	6	NULL	Visa	2025-11-20 07:32:20.7333333	2025-11-20 07:32:20.7333333	29.98	3.90	29.98	NULL	33.88
7	7	NULL	Mastercard	2025-11-19 07:32:20.7333333	2025-11-19 07:32:20.7333333	34.97	4.55	34.97	NULL	39.52
8	NULL	1	CreditCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	25.96	3.37	NULL	25.96	29.33
9	NULL	2	GiftCard	2025-11-21 07:32:20.7333333	2025-11-21 07:32:20.7333333	12.99	1.69	NULL	12.99	14.68
10	NULL	5	CreditCard	2025-11-20 07:32:20.7333333	2025-11-20 07:32:20.7333333	20.97	2.73	NULL	20.97	23.70

CreditCardPayment Table:

Paymen...	CreditCardN...	Amount...
1	4532015112830366	33.88
3	5425233010103442	45.17
5	378282246310005	50.80
6	4916338506082832	33.88
7	5105105105105100	39.52
8	4532015112830366	29.33
10	5425233010103442	23.70

GiftCardPayment Table:

GcTrxnID	Paymen...	GiftCardNum	Amount...	TransactionTime
1	4	9000001234567890	16.94	2025-11-21 07:35:23.7666667
2	9	9000001234567891	14.68	2025-11-21 07:35:23.7666667

SceneRedemption Table:

Redemp...	Paymen...	SceneC...	PointsU...	Amount...	RedemptionTime
1	2	SCENE002	100	22.59	2025-11-21 07:36:35.3233333
2	2	SCENE003	50	5.00	2025-11-21 07:36:35.3233333

Receipt Table:

ReceiptID	Paymen...	UserID	Booking...	SnackOr...	PurchaseDateTime	Total
1	1	1	1	NULL	2025-11-21 07:37:35.9766667	33.88
2	2	2	2	NULL	2025-11-21 07:37:35.9766667	22.59
3	3	3	3	NULL	2025-11-21 07:37:35.9766667	45.17
4	4	4	4	NULL	2025-11-21 07:37:35.9766667	16.94
5	5	5	5	NULL	2025-11-21 07:37:35.9766667	50.80
6	6	6	6	NULL	2025-11-20 07:37:35.9766667	33.88
7	7	7	7	NULL	2025-11-19 07:37:35.9766667	39.52
8	8	1	NULL	1	2025-11-21 07:37:35.9766667	29.33
9	9	2	NULL	2	2025-11-21 07:37:35.9766667	14.68
10	10	5	NULL	5	2025-11-20 07:37:35.9766667	23.70

Sample SQL queries to support applications and reports for management use.

Query 1 - Revenue by Payment Method

```
-- EXPLANATION:
-- This query shows how much revenue was generated from each payment method
-- (Credit Card, Gift Card, Scene+ Points, etc.) over the last 30 days.
-- Useful for Finance team to understand payment trends and customer preferences.
-- Shows: Payment method, transaction count, unique customers, total revenue, and average transaction value.

SELECT
    p.PaymentMethod,
    COUNT(DISTINCT p.PaymentID) AS [Total Transactions],
    COUNT(DISTINCT ISNULL(b.UserID, so.UserID)) AS [Unique Customers],
    SUM(p.TotalPaid) AS [Total Revenue],
    ROUND(AVG(p.TotalPaid), 2) AS [Average Transaction Value]
FROM [Payment] p
LEFT JOIN Booking b ON p.BookingID = b.BookingID
LEFT JOIN SnackOrder so ON p.SnackOrderID = so.SnackOrderID
WHERE p.StartTime >= DATEADD(DAY, -30, GETDATE())
GROUP BY p.PaymentMethod
ORDER BY [Total Revenue] DESC;
```

Paymen...	:	Total Tra...	:	Unique ...	:	Total Re...	:	Average...	:
CreditCard		5		4		182.88		36.580000	
Mastercard		1		1		39.52		39.520000	
Visa		1		1		33.88		33.880000	
GiftCard		2		2		31.62		15.810000	
ScenePlusCard		1		1		22.59		22.590000	

Query 2 - Top 10 Movies by Box Office

```
-- QUERY 2: TOP 10 MOVIES BY BOX OFFICE REVENUE
-- This query identifies the best-performing movies based on ticket sales revenue
-- over the last 30 days. Shows how many screenings, total seats sold, revenue,
-- and average ticket price for each movie.
-- Useful for Programming team to make decisions about which movies to continue showing
-- and which ones to replace. Also helps with marketing strategy for top performers.

SELECT TOP 10
    m.Title,
    m.PG_Rating,
    COUNT(DISTINCT s.ScreeningID) AS [Number of Screenings],
    COUNT(DISTINCT sb.seatBookingID) AS [Total Seats Sold],
    SUM(sb.SeatCost) AS [Box Office Revenue],
    ROUND(AVG(sb.SeatCost), 2) AS [Average Ticket Price]
FROM Movie m
INNER JOIN Screening s ON m.MovieID = s.MovieID
INNER JOIN seatBooking sb ON s.ScreeningID = sb.ScreeningID
WHERE s.ScreeningDate >= DATEADD(DAY, -30, GETDATE())
    AND sb.SeatBookingStatus = 'Booked'
GROUP BY m.MovieID, m.Title, m.PG_Rating
ORDER BY [Box Office Revenue] DESC;
```

Title	PG_Rati...	Number...	Total Se...	Box Offi...	Average...
Interstellar	12A	2	3	59.97	19.990000
Oppenheimer	15	1	2	44.96	22.480000
Inception	12A	1	2	29.98	14.990000
Avatar The Way of Water	12A	1	1	17.49	17.490000
The Dark Knight	12A	1	1	14.99	14.990000
La La Land	12A	1	1	14.99	14.990000

Query 3 - Top 10 Customers by Spending

```
-- QUERY 3: TOP 10 CUSTOMERS BY TOTAL SPENDING
-- This query identifies the most valuable customers based on total spending
-- (both tickets and snacks) over all time. Shows customer name, email, number
-- of visits, spending breakdown, and loyalty membership status.
-- Useful for Marketing team to identify VIP customers for special offers,
-- and for Customer Retention strategies.

SELECT TOP 10
    u.UserID,
    u.[Name],
    u.Email,
    COUNT(DISTINCT b.BookingID) AS [Movie Purchases],
    COUNT(DISTINCT so.SnackOrderID) AS [Snack Orders],
    SUM(b.TotalPrice) AS [Ticket Spending],
    SUM(soi.Subtotal) AS [Snack Spending],
    (SUM(ISNULL(b.TotalPrice, 0)) + SUM(ISNULL(soi.Subtotal, 0))) AS [Total Spending],
    CASE
        WHEN cm.MembershipID IS NOT NULL THEN cm.MembershipType
        ELSE 'Non-Member'
    END AS [Loyalty Status]
FROM [User] u
LEFT JOIN Booking b ON u.UserID = b.UserID
LEFT JOIN SnackOrder so ON u.UserID = so.UserID
LEFT JOIN SnackOrderItem soi ON so.SnackOrderID = soi.SnackOrderID
LEFT JOIN CineClubMembership cm ON u.UserID = cm.UserID
GROUP BY u.UserID, u.[Name], u.Email, cm.MembershipID, cm.MembershipType
HAVING COUNT(DISTINCT b.BookingID) + COUNT(DISTINCT so.SnackOrderID) > 0
ORDER BY [Total Spending] DESC;
```

UserID	Name	Email	Movie P...	Snack O...	Ticket S...	Snack S...	Total Sp...	Loyalty ...
5	Alex Williams	alex.williams@email.com	1	1	89.92	12.98	102.90	Standard
3	Michael Chen	michael.chen@email.com	1	1	79.94	16.48	96.42	Non-Member
1	John Smith	john.smith@email.com	1	1	59.96	25.96	85.92	Standard
6	Jessica Brown	jessica.brown@email.com	1	1	59.96	20.97	80.93	Premium
7	David Martinez	david.martinez@email.com	1	1	34.97	12.99	47.96	Non-Member
2	Sarah Johnson	sarah.johnson@email.com	1	1	19.99	12.99	32.98	Premium
4	Emily Davis	emily.davis@email.com	1	0	14.99	NULL	14.99	VIP

Query 4 - Seat Occupancy by Screening

```
-- QUERY 4: CURRENT SEAT OCCUPANCY BY SCREENING
-- This query shows real-time seat availability for all upcoming screenings.
-- Displays the movie title, theatre complex, screening date/time, total seats,
-- seats booked, seats available, and occupancy percentage.
-- Useful for Operations team to monitor occupancy rates and identify
-- which screenings are selling well and which may need promotion.

SELECT
    m.Title,
    c.[Name] AS [Theatre Complex],
    h.HallNumber,
    s.ScreeningDate,
    s.StartTime,
    h.SeatCapacity,
    ISNULL(COUNT(DISTINCT sb.seatBookingID), 0) AS [Seats Booked],
    (h.SeatCapacity - ISNULL(COUNT(DISTINCT sb.seatBookingID), 0)) AS [Seats Available],
    ROUND(
        (ISNULL(COUNT(DISTINCT sb.seatBookingID), 0) * 100.0 / h.SeatCapacity), 2
    ) AS [Occupancy Percentage]
FROM Screening s
INNER JOIN Movie m ON s.MovieID = m.MovieID
INNER JOIN Hall h ON s.HallID = h.HallID
INNER JOIN [Complex] c ON h.ComplexID = c.ComplexID
LEFT JOIN seatBooking sb ON s.ScreeningID = sb.ScreeningID
    AND sb.SeatBookingStatus = 'Booked'
WHERE s.ScreeningDate >= GETDATE()
GROUP BY m.MovieID, m.Title, c.ComplexID, c.[Name], h.HallID, h.HallNumber,
    h.SeatCapacity, s.ScreeningID, s.ScreeningDate, s.StartTime
ORDER BY s.ScreeningDate, s.StartTime;
```

Title	Theatre Complex	HallNu...	Screeni...	StartTime	SeatCap...	Seats B...	Seats Av...	Occupan...
Oppenheimer	Cineplex Odeon Scotiabank Theatre	1	2025-11-22	10:00:00	250	2	248	0.800000000000
The Shawshank Redemption	Cineplex Odeon Scotiabank Theatre	2	2025-11-22	14:00:00	180	0	180	0.000000000000
The Dark Knight	Cineplex Odeon Forum Theatre	1	2025-11-22	16:00:00	120	1	119	0.830000000000
Dune Part Two	Cineplex Odeon Yonge-Dundas	1	2025-11-22	18:00:00	200	0	200	0.000000000000
Avatar The Way of Water	Cineplex Odeon Mississauga	1	2025-11-23	15:30:00	200	1	199	0.500000000000

Query 5 - Top 10 Snacks by Revenue

```
-- QUERY 5: TOP 10 SNACKS BY SALES REVENUE
-- This query shows which concession items generate the most revenue
-- over the last 30 days. Displays snack name, category, units sold,
-- total revenue, and average price per unit.
-- Useful for Concessions Manager to understand which products are best-sellers,
-- manage inventory, and determine pricing strategy for promotions.
```

```
SELECT TOP 10
    sn.SnackName,
    sn.Category,
    ROUND(sn.Price, 2) AS [Unit Price],
    SUM(soi.Quantity) AS [Total Quantity Sold],
    SUM(soi.Subtotal) AS [Total Revenue],
    ROUND(AVG(soi.PricePerUnit), 2) AS [Average Price Paid]
FROM Snack sn
INNER JOIN SnackOrderItem soi ON sn.SnackID = soi.SnackID
WHERE soi.SnackOrderID IN
    (SELECT SnackOrderID FROM SnackOrder
     WHERE OrderTime >= DATEADD(DAY, -30, GETDATE()))
GROUP BY sn.SnackID, sn.SnackName, sn.Category, sn.Price
ORDER BY [Total Revenue] DESC;
```

SnackName	Category	Unit Price	Total Qu...	Total Re...	Average...
Combo - Popcorn + Drink	Combo	12.99	2	25.98	12.990000
Small Popcorn	Popcorn	6.99	3	20.97	6.990000
Small Coca-Cola	Beverage	5.99	3	17.97	5.990000
Gummy Bears	Candy	5.49	2	10.98	5.490000
Large Popcorn	Popcorn	9.99	1	9.99	9.990000
Buttered Popcorn	Popcorn	8.49	1	8.49	8.490000
Large Coca-Cola	Beverage	7.99	1	7.99	7.990000

Evaluation and Conclusion

The Achievement of the Project

Our project successfully designed and implemented a comprehensive relational database system for Cineplex, capturing the full complexity of cinema operations including user management, bookings, payments, loyalty, and sales analytics. Our team produced a normalized, scalable schema with 29 interrelated tables, ensuring strong data integrity and referential consistency. Key deliverables included robust table creation scripts, full sample data population, and business-focused SQL queries for real-time management reporting. The project meets, course requirements by providing both technical correctness and practical application value, supporting all essential business processes for a large multi-theatre cinema chain.

Further Development and Improvement

Future enhancements can focus on stored procedures for transaction workflows such as booking, payment, and receipt issuance, as well as implementation of database views for simplified reporting. Introducing advanced analytics, such as customer segmentation and dynamic ticket pricing, would add strategic value. Security could be elevated with user roles and audit logging. In the long term, separating transactional and analytical workloads into OLAP and OLTP systems, plus adding support for mobile and multi-language functionality, would make the system more robust and adaptable for potential expansion or integration with other business platforms.

Advantages and Limitations of the Database Package Used

Microsoft SQL Server offers robust support for relational integrity, advanced data types, indexing, transaction management, and compatibility with industry-standard BI tools. Its integrated development environment simplifies schema creation, debugging, and backup. However, SQL Server restricts multiple cascading actions across relationships, occasionally requiring workaround designs (e.g., ON DELETE NO ACTION), and its licensing costs may not be ideal for small-scale or open-source projects. Performance can also be resource-intensive as data volumes grow. Despite these limitations, it remains a reliable, well-documented choice for medium to large enterprise database solutions.

The Experience Learned from This Project

This project provided us with hands-on experience of every phase of database development: conceptual modeling, normalization, schema implementation, integrity enforcement, and real-world SQL coding. Dealing with relationship complexity and constraint management in SQL Server reinforced the importance of proper planning, testing, and clear documentation. Collaborative teamwork allowed for effective role

distribution and mutual feedback. The process also highlighted the need for iterative design, resolving issues such as foreign key loops and identity constraints refined both technical skills and project management competencies, equipping the team for future, larger-scale database projects.

Contribution by Each Team Member

1. Mohammad Areeb (220094736)

- Led database design, normalization, and schema creation.
- Developed and refined the ER model.
- Implemented all SQL table scripts, sample data, queries, and integrity controls.
- Authored the complete project proposal and progress report.
- Coordinated final report writing and resolved major design issues.

2. Prabhkrit Singh (220164059)

- Supported ER model development and refinement.
- Assisted with report drafting and ERD documentation.
- Provided feedback on schema and relationships.

3. Samarjeet Singh Arora (220857686)

- Participated in ER model drafting.
- Assisted with attribute validation and relationship mapping.

4. Akash Deep (220854162)

- Helped in initial ER model design, relationships, and attribute definition.
- Supported final report preparation and conceptual design.

5. Karthikeya Gorijavolu (220241030)

- Wrote background and objectives.
- Provided business context.

Note:

- All members contributed to team meetings and review sessions.
- Key technical leadership, major documentation (including proposal and progress report), and overall project integration were led by Mohammad Areeb, supported by effective collaboration from all members.