CSCI 327 – Project: Deliverable 2

Group Members

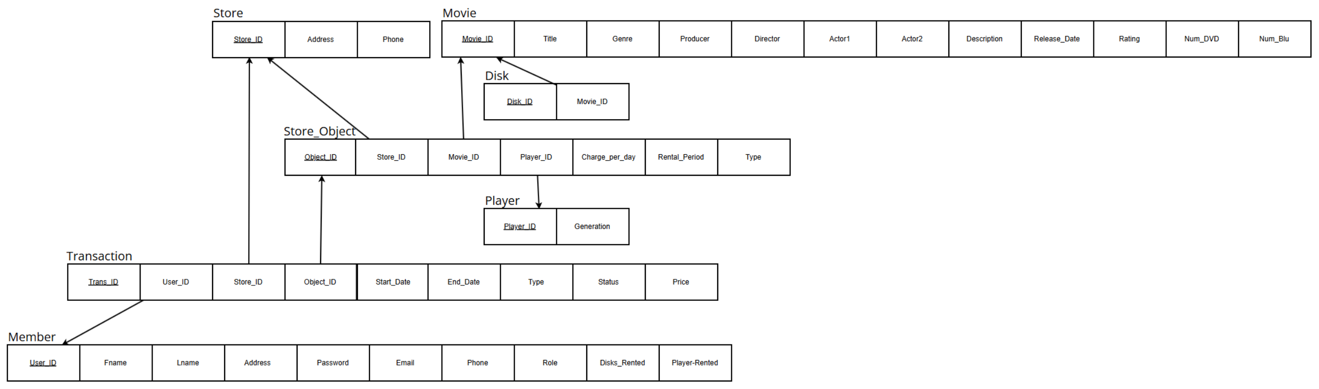
Our team consists of *two* students:

- **Sepehr Akbari**
- **Ilana Berlin**

Goals

1. Creating the Relational Diagram
2. Making the Table using SQL
3. Inserting Data into Tables using SQL (the population is done to test functionality later)
4. Outlining some difficulties we faced during the process

Relational Mapping



Approach:

- All entities (Member, Store, Store_Object, Movie, Disk, and Player) becomes a table.
- Ternary relationship Transaction is M:N:P and has its own attributes so it becomes its own table with foreign keys “Store_ID”, ‘Object_ID”, and “UserID”.
- Binary relationship Owns is 1:N so a foreign key “Store_ID” is added to Store_Object.
- Binary relationship is_on is 1:N so foreign key is “Movie_ID” is added to Disk.
- Superclass/subclass relationship between parent class Store_Object and child classes Movie and Player mapped using Option C.
- Each Table gets an ID as a primary key.

Creating the Table Schema and Populating

SQL Code for Creating Tables

```
-- RESET
DROP DATABASE IF EXISTS VideoStore;
CREATE DATABASE VideoStore;

-- USING
USE VideoStore;

-- STORE TABLE
CREATE TABLE Store (
    Store_ID INT PRIMARY KEY AUTO_INCREMENT,
    Address VARCHAR(255) UNIQUE NOT NULL,
    Phone VARCHAR(15) UNIQUE NOT NULL
);

-- MEMBER TABLE
CREATE TABLE Member (
    UserID INT PRIMARY KEY AUTO_INCREMENT,
    Fname VARCHAR(100) NOT NULL,
    Lname VARCHAR(100) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Password VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    Phone VARCHAR(15) UNIQUE,
    Role VARCHAR(20) NOT NULL CHECK (Role IN ('Customer', 'Admin')),
    Num_disks_rented INT DEFAULT 0 CHECK (Num_disks_rented <= 10),
    Player_rented INT DEFAULT 0 CHECK (Player_rented <= 1)
);

-- MOVIE TABLE
CREATE TABLE Movie (
    Movie_ID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(255) NOT NULL,
    Genre VARCHAR(50),
    Producer VARCHAR(100),
    Director VARCHAR(100),
    Actor1 VARCHAR(100),
    Actor2 VARCHAR(100),
    Description TEXT,
    Release_date DATE,
    Rating DECIMAL(2, 1) CHECK (Rating >= 0 AND Rating <= 10),
    Num_DVD INT DEFAULT 1,
    Num_Blu INT DEFAULT 0,
    CHECK (Num_DVD >= 0 OR Num_Blu >= 0)
);

-- PLAYER TABLE
CREATE TABLE Player (
    Player_ID INT PRIMARY KEY AUTO_INCREMENT,
    Generation INT DEFAULT 2,
    CHECK (Generation >= 1 AND Generation <= 3)
);
```

```sql
-- STORE-OBJECT TABLE
CREATE TABLE Store_Object (
    Object_ID INT PRIMARY KEY AUTO_INCREMENT,
    Store_ID INT NOT NULL,
    Movie_ID INT,
    Player_ID INT,
    Charge_per_day DECIMAL(5, 2) NOT NULL,
    Rental_period INT DEFAULT 3,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('DVD', 'Blu-Ray')),
    FOREIGN KEY (Store_ID) REFERENCES Store(Store_ID),
    FOREIGN KEY (Movie_ID) REFERENCES Movie(Movie_ID),
    FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID),
    CHECK(
        (Movie_ID IS NOT NULL AND Player_ID IS NULL) OR
        (Movie_ID IS NULL AND Player_ID IS NOT NULL)
    ),
    CHECK (Rental_period >= 1)
);

-- TRANSACTION TABLE
CREATE TABLE Transaction (
    Trans_ID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT NOT NULL,
    Object_ID INT NOT NULL,
    Store_ID INT NOT NULL,
    Start_date DATE,
    End_date DATE,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('Rental', 'Reservation')),
    Status VARCHAR(20) NOT NULL CHECK (Status IN ('Active', 'Completed',
'Reserved')),
    Price DECIMAL(10, 2),
    CHECK (End_date >= Start_date),
    CHECK (
        (Status = 'Completed' AND End_date IS NOT NULL) OR
        (Status <> 'Completed' AND End_date IS NULL)
    ),
    CHECK (
        (Start_date is NULL AND Status = 'Reserved') OR
        (Start_date IS NOT NULL AND Status <> 'Reserved')
    ),
    CHECK (
        (End_date IS NULL AND Price IS NULL) OR
        (End_date IS NOT NULL AND Price IS NOT NULL)
    ),
    FOREIGN KEY (UserID) REFERENCES Member(UserID),
    FOREIGN KEY (Object_ID) REFERENCES Store_Object(Object_ID),
    FOREIGN Key (Store_ID) REFERENCES Store (Store_ID)
);

-- DISK TABLE
CREATE TABLE Disk (
    Disk_ID INT PRIMARY KEY AUTO_INCREMENT,
    Movie_ID INT NOT NULL,
    FOREIGN KEY (Movie_ID) REFERENCES Movie(Movie_ID)
);

-- INDEXES
CREATE INDEX idx_movie_title ON Movie(Title);
CREATE INDEX idx_movie_genre ON Movie(Genre);
CREATE INDEX idx_movie_director ON Movie(Director);
CREATE INDEX idx_transaction_userid ON Transaction(UserID);
```

SQL Code for Populating Tables

```
USE VideoStore;

-- STORE TABLE
INSERT INTO Store (Address, Phone) VALUES
('Lake Forest College, Brown Hall', '555-1234'),
('Lake Forest College, Lillard Hall', '555-5678'),
('Lake Forest College Mohr Center', '555-9101'),
('Lake Forest College, North Campus', '555-1121'),
('Lake Forest College, South Campus', '555-3141');

-- MEMBER TABLE
INSERT INTO Member (Fname, Lname, Address, Password, Email, Phone, Role,
Num_disks_rented, Player_rented) VALUES
('Sepehr', 'Akbari', 'Harlan Hall, 224', 'sepehr123', 'sepehr@lfc.edu',
'999-111', 'Admin', 0, 0),
('Ilana', 'Berlin', 'Nollen Hall, 021', 'ilana123', 'ilana@lfc.edu', '999-
222', 'Admin', 0, 0),
('Adela', 'Bragg', 'Nollen Hall, 021', 'adela123', 'adela@lfc.edu', '888-
001', 'Customer', 0, 0),
('Adam', 'Kaderbhai', 'Deerpath Hall, 017', 'adam123', 'adam@lfc.edu',
'888-002', 'Customer', 0, 0),
('Zach', 'Buhai', 'Deerpath Hall, 019', 'zach123', 'zach@lfc.edu', '888-
003', 'Customer', 8, 1),
('Sugata', 'Banerji', 'Brown Hall, 254', 'sugata123', 'sugata@lfc.edu',
'888-004', 'Customer', 3, 0);

-- MOVIE TABLE
INSERT INTO Movie (Title, Genre, Producer, Director, Actor1, Actor2,
Description, Release_date, Rating, Num_DVD, Num_Blu) VALUES
('The Shawshank Redemption', 'Drama', 'Niki Marvin', 'Frank Darabont',
'Tim Robbins', 'Morgan Freeman', 'Two imprisoned men bond over a number of
years, finding solace and eventual redemption through acts of common
decency.', '1994-09-23', 9.3, 1, 0),
('The Godfather', 'Crime', 'Albert S. Ruddy', 'Francis Ford Coppola',
'Marlon Brando', 'Al Pacino', 'The aging patriarch of an organized crime
dynasty transfers control of his clandestine empire to his reluctant
son.', '1972-03-24', 9.2, 1, 0),
('Inception', 'Sci-Fi', 'Christopher Nolan', 'Christopher Nolan',
'Leonardo DiCaprio', 'Joseph Gordon-Levitt', 'A thief who steals corporate
secrets through dream infiltration is given the task of planting an idea
into the mind of a CEO.', '2010-07-16', 8.8, 1, 0),
('Pulp Fiction', 'Crime', 'Lawrence Bender', 'Quentin Tarantino', 'John
Travolta', 'Samuel L. Jackson', 'The lives of two mob hitmen, a boxer, a
gangster and his wife, and a pair of diner bandits intertwine in four
tales of violence and redemption.', '1994-10-14', 8.9, 1, 0),
('Forrest Gump', 'Drama', 'Wendy Finerman', 'Robert Zemeckis', 'Tom
Hanks', 'Robin Wright', 'The presidencies of Kennedy and Johnson, the
Vietnam War, and other events unfold through the perspective of an Alabama
man with an IQ of 75.', '1994-07-06', 8.8, 1, 0),
('The Dark Knight', 'Action', 'Christopher Nolan', 'Christopher Nolan',
'Christian Bale', 'Heath Ledger', 'When the menace known as the Joker
emerges, Batman must accept one of the greatest psychological and physical
tests of his ability to fight injustice.', '2008-07-18', 9.0, 1, 0),
('Titanic', 'Romance', 'James Cameron', 'James Cameron', 'Leonardo
DiCaprio', 'Kate Winslet', 'A seventeen-year-old aristocrat falls in love
with a kind but poor artist aboard the luxurious, ill-fated R.M.S.
Titanic.', '1997-12-19', 7.8, 1, 0),
('The Matrix', 'Sci-Fi', 'Joel Silver', 'Lana Wachowski', 'Keanu Reeves',
'Laurence Fishburne', 'A computer hacker learns from mysterious rebels
about the true nature of his reality and his role in the war against its
controllers.', '1999-03-31', 8.7, 1, 0),
('Gladiator', 'Action', 'David Franzoni', 'Ridley Scott', 'Russell Crowe',
'Joaquin Phoenix', 'A former Roman General sets out to exact vengeance
against the corrupt emperor who murdered his family and sent him into
slavery.', '2000-05-05', 8.5, 1, 0),
('La La Land', 'Romance', 'Fred Berger', 'Damien Chazelle', 'Ryan
Gosling', 'Emma Stone', 'While navigating their careers in Los Angeles, a
pianist and an actress fall in love while attempting to reconcile their
aspirations for the future.', '2016-12-09', 8.0, 1, 0),
('The Silence of the Lambs', 'Thriller', 'Edward Saxon', 'Jonathan Demme',
'Jodie Foster', 'Anthony Hopkins', 'A young FBI cadet must receive the
help of an incarcerated and manipulative cannibal killer to help catch
another serial killer.', '1991-02-14', 8.6, 1, 0),
('Jurassic Park', 'Adventure', 'Kathleen Kennedy', 'Steven Spielberg',
'Sam Neill', 'Laura Dern', 'A pragmatic paleontologist touring an almost
complete theme park is tasked with protecting a couple of kids after a
power failure causes the park''s cloned dinosaurs to run loose.', '1993-
06-11', 8.1, 1, 0),
('Avatar', 'Sci-Fi', 'James Cameron', 'James Cameron', 'Sam Worthington',
'Zoe Saldana', 'A paraplegic Marine dispatched to the moon Pandora on a
unique mission becomes torn between following his orders and protecting
the world he feels is his home.', '2009-12-18', 7.8, 1, 0),
('The Lion King', 'Animation', 'Don Hahn', 'Roger Allers', 'Matthew
Broderick', 'Jeremy Irons', 'Lion prince Simba and his father are targeted
by his bitter uncle, who wants to ascend the throne himself.', '1994-06-
24', 8.5, 1, 0),
('Schindler''s List', 'History', 'Steven Spielberg', 'Steven Spielberg',
'Liam Neeson', 'Ben Kingsley', 'In German-occupied Poland during World War
II, industrialist Oskar Schindler gradually becomes concerned for his
Jewish workforce after witnessing their persecution by the Nazis.', '1993-
12-15', 8.9, 1, 0);

-- PLAYER TABLE
INSERT INTO Player (Generation) VALUES
(1), -- Player 1: 1st generation
```

```sql
(2), -- Player 2: 2nd generation
(2), -- Player 3: 2nd generation
(3), -- Player 4: 3rd generation
(3); -- Player 5: 3rd generation

-- STORE_OBJECT TABLE
INSERT INTO Store_Object (Store_ID, Movie_ID, Player_ID, Charge_per_day,
Rental_period, Type) VALUES
(1, 1, NULL, 2.50, 3, 'DVD'), (1, 1, NULL, 2.50, 3, 'DVD'), (2, 1, NULL,
3.00, 3, 'Blu-Ray'), -- The Shawshank Redemption: 2 DVDs, 1 Blu-Ray
(1, 2, NULL, 2.00, 3, 'DVD'), (2, 2, NULL, 2.00, 3, 'DVD'), -- The
Godfather: 2 DVDs
(2, 3, NULL, 3.00, 3, 'Blu-Ray'), (3, 3, NULL, 2.50, 3, 'DVD'), (3, 3,
NULL, 2.50, 3, 'DVD'), -- Inception: 1 Blu-Ray, 2 DVDs
(1, 4, NULL, 2.50, 3, 'DVD'), (3, 4, NULL, 2.75, 3, 'Blu-Ray'), (3, 4,
NULL, 2.75, 3, 'Blu-Ray'), -- Pulp Fiction: 1 DVD, 2 Blu-Ray
(2, 5, NULL, 1.50, 3, 'DVD'), (3, 5, NULL, 1.50, 3, 'DVD'), -- Forrest
Gump: 2 DVDs
(1, 6, NULL, 3.50, 5, 'Blu-Ray'), (2, 6, NULL, 3.00, 3, 'DVD'), (3, 6,
NULL, 3.00, 3, 'DVD'), -- The Dark Knight: 1 Blu-Ray, 2 DVDs
(1, 7, NULL, 2.00, 3, 'DVD'), (2, 7, NULL, 2.00, 3, 'DVD'), -- Titanic: 2
DVDs
(1, 8, NULL, 2.75, 3, 'Blu-Ray'), (3, 8, NULL, 2.50, 3, 'DVD'), (3, 8,
NULL, 2.50, 3, 'DVD'), -- The Matrix: 1 Blu-Ray, 2 DVDs
(2, 9, NULL, 2.25, 3, 'DVD'), (3, 9, NULL, 2.25, 3, 'DVD'), -- Gladiator:
2 DVDs
(1, 10, NULL, 3.00, 3, 'Blu-Ray'), (2, 10, NULL, 2.75, 3, 'DVD'), (3, 10,
NULL, 2.75, 3, 'DVD'), -- La La Land: 1 Blu-Ray, 2 DVDs
(1, 11, NULL, 2.50, 3, 'DVD'), (2, 11, NULL, 2.50, 3, 'DVD'), -- The
Silence of the Lambs: 2 DVDs
(1, 12, NULL, 2.00, 3, 'DVD'), (3, 12, NULL, 2.00, 3, 'DVD'), -- Jurassic
Park: 2 DVDs
(2, 13, NULL, 3.25, 3, 'Blu-Ray'), (3, 13, NULL, 3.00, 3, 'DVD'), (3, 13,
NULL, 3.00, 3, 'DVD'), -- Avatar: 1 Blu-Ray, 2 DVDs
(1, 14, NULL, 1.75, 3, 'DVD'), (2, 14, NULL, 1.75, 3, 'DVD'), -- The Lion
King: 2 DVDs
(1, 15, NULL, 3.00, 3, 'Blu-Ray'), (1, 15, NULL, 2.75, 3, 'DVD'), (3, 15,
NULL, 2.75, 3, 'DVD'), -- Schindler's List: 1 Blu-Ray, 2 DVDs
(1, NULL, 1, 5.00, 7, 'DVD'), -- Player 1 (Store 1)
(1, NULL, 2, 5.00, 7, 'DVD'), -- Player 2 (Store 1)
(2, NULL, 3, 5.00, 7, 'DVD'), -- Player 3 (Store 2)
(2, NULL, 4, 5.00, 7, 'DVD'), -- Player 4 (Store 2)
(3, NULL, 5, 5.00, 7, 'DVD'); -- Player 5 (Store 3)

-- Updating Movie
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 1 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 1 AND Type = 'Blu-Ray') WHERE Movie_ID = 1;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 2 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 2 AND Type = 'Blu-Ray') WHERE Movie_ID = 2;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 3 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 3 AND Type = 'Blu-Ray') WHERE Movie_ID = 3;
```

```
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 4 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 4 AND Type = 'Blu-Ray') WHERE Movie_ID = 4;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 5 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 5 AND Type = 'Blu-Ray') WHERE Movie_ID = 5;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 6 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 6 AND Type = 'Blu-Ray') WHERE Movie_ID = 6;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 7 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 7 AND Type = 'Blu-Ray') WHERE Movie_ID = 7;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 8 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 8 AND Type = 'Blu-Ray') WHERE Movie_ID = 8;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 9 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 9 AND Type = 'Blu-Ray') WHERE Movie_ID = 9;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 10 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 10 AND Type = 'Blu-Ray') WHERE Movie_ID =
10;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 11 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 11 AND Type = 'Blu-Ray') WHERE Movie_ID =
11;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 12 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 12 AND Type = 'Blu-Ray') WHERE Movie_ID =
12;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 13 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 13 AND Type = 'Blu-Ray') WHERE Movie_ID =
13;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 14 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 14 AND Type = 'Blu-Ray') WHERE Movie_ID =
14;
UPDATE Movie SET Num_DVD = (SELECT COUNT(*) FROM Store_Object WHERE
Movie_ID = 15 AND Type = 'DVD'), Num_Blu = (SELECT COUNT(*) FROM
Store_Object WHERE Movie_ID = 15 AND Type = 'Blu-Ray') WHERE Movie_ID =
15;

-- DISK TABLE
INSERT INTO Disk (Movie_ID) VALUES
(1), (1), (1), (1), -- The Shawshank Redemption: 4 disks (3 copies)
(2), (2), (2), -- The Godfather: 3 disks (2 copies)
(3), (3), (3), (3), -- Inception: 4 disks (3 copies)
(4), (4), (4), (4), -- Pulp Fiction: 4 disks (3 copies)
(5), (5), (5), -- Forrest Gump: 3 disks (2 copies)
(6), (6), (6), (6), -- The Dark Knight: 4 disks (3 copies)
(7), (7), (7), -- Titanic: 3 disks (2 copies)
(8), (8), (8), (8), -- The Matrix: 4 disks (3 copies)
(9), (9), (9), -- Gladiator: 3 disks (2 copies)
```

```
(10), (10), (10), (10), -- La La Land: 4 disks (3 copies)
(11), (11), (11), -- The Silence of the Lambs: 3 disks (2 copies)
(12), (12), (12), -- Jurassic Park: 3 disks (2 copies)
(13), (13), (13), (13), -- Avatar: 4 disks (3 copies)
(14), (14), (14), -- The Lion King: 3 disks (2 copies)
(15), (15), (15), (15); -- Schindler's List: 4 disks (3 copies)

-- TRANSACTION TABLE
INSERT INTO Transaction (UserID, Object_ID, Store_ID, Start_date,
End_date, Type, Status, Price) VALUES
(3, 3, 2, '2025-03-01', '2025-03-05', 'Rental', 'Completed', 12.00), --
Adela rents Inception Blu-Ray (4 days * $3.00)
(3, 6, 2, '2025-03-10', NULL, 'Rental', 'Active', NULL), -- Adela rents
The Dark Knight Blu-Ray (active)
(4, 10, 3, NULL, NULL, 'Reservation', 'Reserved', NULL), -- Adam reserves
Pulp Fiction Blu-Ray
(4, 17, 1, '2025-03-05', '2025-03-08', 'Rental', 'Completed', 6.00), --
Adam rents Titanic DVD (3 days * $2.00)
(5, 14, 1, '2025-03-03', NULL, 'Rental', 'Active', NULL), -- Zach rents
The Dark Knight Blu-Ray (active)
(5, 19, 1, '2025-03-01', '2025-03-04', 'Rental', 'Completed', 8.25), --
Zach rents The Matrix Blu-Ray (3 days * $2.75)
(5, 24, 1, '2025-03-05', NULL, 'Rental', 'Active', NULL), -- Zach rents La
La Land Blu-Ray (active)
(5, 27, 1, NULL, NULL, 'Reservation', 'Reserved', NULL), -- Zach reserves
The Silence of the Lambs DVD
(5, 31, 2, '2025-03-04', NULL, 'Rental', 'Active', NULL), -- Zach rents
Avatar Blu-Ray (active)
(5, 36, 1, '2025-03-01', '2025-03-06', 'Rental', 'Completed', 15.00), --
Zach rents Schindler's List Blu-Ray (5 days * $3.00)
(6, 12, 2, '2025-03-03', '2025-03-05', 'Rental', 'Completed', 3.00), --
Sugata rents Forrest Gump DVD (2 days * $1.50)
(6, 16, 2, '2025-03-06', NULL, 'Rental', 'Active', NULL), -- Sugata rents
The Dark Knight DVD (active)
(6, 7, 3, '2025-03-02', '2025-03-04', 'Rental', 'Completed', 5.00), --
Sugata rents Inception DVD (2 days * $2.50)
(5, 41, 1, '2025-03-05', NULL, 'Rental', 'Active', NULL), -- Zach rents
Player 1 (active)
(5, 8, 3, '2025-03-01', '2025-03-03', 'Rental', 'Completed', 5.00), --
Zach rents Inception DVD (2 days * $2.50)
(5, 22, 2, NULL, NULL, 'Reservation', 'Reserved', NULL), -- Zach reserves
Gladiator DVD
(5, 28, 1, '2025-03-06', NULL, 'Rental', 'Active', NULL), -- Zach rents
The Silence of the Lambs DVD (active)
(4, 29, 1, '2025-03-03', '2025-03-05', 'Rental', 'Completed', 4.00), --
Adam rents Jurassic Park DVD (2 days * $2.00)
(3, 32, 3, '2025-03-02', NULL, 'Rental', 'Active', NULL), -- Adela rents
Avatar DVD (active)
(5, 43, 2, '2025-03-01', '2025-03-08', 'Rental', 'Completed', 35.00); --
Zach rents Player 3 (7 days * $5.00)

-- Updating Member
UPDATE Member SET Num_disks_rented = 0, Player_rented = 0 WHERE UserID =
1; -- Sepehr (Admin)
```

```
UPDATE Member SET Num_disks_rented = 0, Player_rented = 0 WHERE UserID =
2; -- Ilana (Admin)
UPDATE Member SET Num_disks_rented = 2, Player_rented = 0 WHERE UserID =
3; -- Adela: 2 active movie rentals
UPDATE Member SET Num_disks_rented = 0, Player_rented = 0 WHERE UserID =
4; -- Adam: No active rentals
UPDATE Member SET Num_disks_rented = 5, Player_rented = 1 WHERE UserID =
5; -- Zach: 5 active movie rentals, 1 player
UPDATE Member SET Num_disks_rented = 1, Player_rented = 0 WHERE UserID =
6; -- Sugata: 1 active movie rental
```

Difficulties

- Determining what needed to be included in the deliverable
- Relating Disk and Player to Store_Object during table creation.
- Calculating the total price of a transaction since start and end dates are in a different table from price per day.
- Started with an overcomplicated schema that we ended up simplifying, removed separate tables from director and producer, and changed actor from a multivalued attribute.
- Combined Reservation and Rental into Transaction
- We first started with player_rented as a boolean but then changed it to int.
- We started with Type of Disk as a boolean and then changed it to a Char
- We ran into chicken and egg problems with foreign keys while creating the tables and inserting data into the tables
- We added the following checks to make sure illogical data is not entered:
    - Num_disks_rented <= 10
    - Player_rented <= 1
    - Rating >= 0 AND Rating <= 10
    - Num_DVD >= 0 OR Num_Blu >= 0
    - Generation >= 1 AND Generation <= 3
    - Rental_period >= 1
    - Start_date is NULL AND Status = 'Reserved' OR Start_date IS NOT NULL AND Status <> 'Reserved'
    - End_date IS NULL AND Price IS NULL OR End_date IS NOT NULL AND Price IS NOT NULL
    - Status = 'Completed' AND End_date IS NOT NULL OR Status <> 'Completed' AND End_date IS NULL