QUERIES:

```sql
--Q1 retrieves the total price of reservations made by a loyalty customer named Eslla
Hall  (3 Tables)
SELECT L.CustomerName, SUM(P.Price) AS TotalPrice
FROM LoyaltyCustomers L
JOIN Reservations R ON L.CustomerId = R.CustomerId
JOIN Prices P ON R.MovieId = P.MovieId
GROUP BY L.CustomerName
HAVING L.CustomerName = 'Ella Hall';
--The JOIN clause is used to combine data from the LoyaltyCustomers, Reservations, and
Prices tables based on the respective IDs.
--The GROUP BY clause groups the rows based on the CustomerName.
--The SUM(P.Price) calculates the total price of reservations for each customer.
--The HAVING clause filters the results to include only the rows where the CustomerName
is 'Ella Hall'.


--Q2 loyalty customers who have made reservations for a specific movie with the details
of their reservations
SELECT L.CustomerName, L.Email, M.MovieTitle, R.ReservationDate, R.ReservationTime,
S.SeatNumber
FROM LoyaltyCustomers L
JOIN Reservations R ON L.CustomerId = R.CustomerId
JOIN Movies M ON R.MovieId = M.MovieId
JOIN Seats S ON R.SeatId = S.SeatId
WHERE M.MovieTitle = 'Dilwale Dulhania Le Jayenge'
ORDER BY R.ReservationDate DESC, R.ReservationTime DESC;
--The WHERE clause filters the results to include only the rows where the MovieTitle is
'Dilwale Dulhania Le Jayenge


--Q3:reservation details along with the movie title, customer name, and branch address
(6 tables)
SELECT Reservations.ReservationId, Movies.MovieTitle, LoyaltyCustomers.CustomerName,
Branches.Address
FROM Reservations
JOIN Movies ON Reservations.MovieId = Movies.MovieId
JOIN LoyaltyCustomers ON Reservations.CustomerId = LoyaltyCustomers.CustomerId
JOIN Seats ON Reservations.SeatId = Seats.SeatId
JOIN Rooms ON Seats.RoomId = Rooms.RoomId
JOIN Branches ON Rooms.BranchID = Branches.BranchID;


--Q4 retrieve all movies and their corresponding branches  (3 tables)
SELECT Movies.MovieTitle, Branches.BranchName
FROM Movies
INNER JOIN Rooms ON Movies.MovieId = Rooms.MovieId
INNER JOIN Branches ON Rooms.BranchID = Branches.BranchID;


--Q5 retrieve total number of reservations made by each customer at a specific branch (5
tables)
```

```sql
SELECT LoyaltyCustomers.CustomerName, COUNT(*) AS ReservationCount
FROM Reservations
JOIN LoyaltyCustomers ON Reservations.CustomerId = LoyaltyCustomers.CustomerId
JOIN Seats ON Reservations.SeatId = Seats.SeatId
JOIN Rooms ON Seats.RoomId = Rooms.RoomId
JOIN Branches ON Rooms.BranchID = Branches.BranchID
WHERE Branches.BranchName = 'Centaurus Mall'
GROUP BY LoyaltyCustomers.CustomerName;

--Q6  retrieve info about employees, their assigned branches, their shifts, and the rooms associated with those branches
SELECT E.EmployeeName, B.BranchName, S.ShiftTime, R.RoomNumber
FROM Employees E
INNER JOIN Branches B ON E.BranchID = B.BranchID
INNER JOIN EmployeeShifts ES ON E.EmployeeId = ES.EmployeeId
INNER JOIN Shifts S ON ES.ShiftId = S.ShiftId
INNER JOIN Rooms R ON B.BranchID = R.BranchID;


--Q7  retrieves the top 3 movies with the highest number of reservations in desc order
SELECT TOP 3 M.MovieTitle, COUNT(R.ReservationId) AS ReservationCount
FROM Movies M
JOIN Reservations R ON M.MovieId = R.MovieId
JOIN Seats S ON R.SeatId = S.SeatId
JOIN Rooms RM ON S.RoomId = RM.RoomId
GROUP BY M.MovieTitle
ORDER BY ReservationCount DESC;


--Q8 retrieve the information about a specific customer and the branch where they have watched which movie
SELECT C.CustomerName, B.BranchName, M.MovieTitle
FROM LoyaltyCustomers C
JOIN Reservations R ON C.CustomerId = R.CustomerId
JOIN Seats S ON R.SeatId = S.SeatId
JOIN Rooms RM ON S.RoomId = RM.RoomId
JOIN Branches B ON RM.BranchID = B.BranchID
JOIN Movies M ON R.MovieId = M.MovieId
WHERE C.CustomerName = 'Noah Wilson';


--Q9 retrieves the customer name, movie title, and the food they have ordered of all the customers
SELECT L.CustomerName, M.MovieTitle, F.FoodName
FROM LoyaltyCustomers L
JOIN Reservations R ON L.CustomerId = R.CustomerId
JOIN Movies M ON R.MovieId = M.MovieId
JOIN Seats S ON R.SeatId = S.SeatId
JOIN Rooms RM ON S.RoomId = RM.RoomId
JOIN Food F ON RM.BranchID = F.BranchID
--WHERE LC.Custom


-- Q10 find the total amount generated by selling food in each branch for confirmed reservations
SELECT b.BranchName, SUM(f.Price) AS TotalAmount
FROM Branches b
JOIN Rooms r ON b.BranchID = r.BranchID
```

```sql
JOIN Seats s ON r.RoomId = s.RoomId
JOIN Reservations res ON s.SeatId = res.SeatId
JOIN Food f ON b.BranchID = f.BranchID
WHERE res.ReservationDate IS NOT NULL -- Consider only confirmed reservations
GROUP BY b.BranchName;


-- Q11 Retrieve the list of movies currently scheduled to be shown in a specific branch -
Packages Mall
SELECT DISTINCT m.MovieTitle, m.Director, m.Genre, m.Duration, m.ReleaseDate
FROM Movies m
JOIN Schedule s ON m.MovieId = s.MovieId
JOIN Rooms r ON s.RoomId = r.RoomId
JOIN Branches b ON r.BranchID = b.BranchID
WHERE b.BranchName = 'Packages Mall';


--Q12 Calculate the total revenue generated by ticket sales for each movie
SELECT m.MovieTitle, SUM(p.Price) AS TotalRevenue
FROM Movies m
JOIN Prices p ON m.MovieId = p.MovieId
JOIN Reservations res ON m.MovieId = res.MovieId
GROUP BY m.MovieTitle;


--Q13 Find the most popular movie genre based on the number of reservations
SELECT TOP 1 m.Genre, COUNT(res.ReservationId) AS ReservationCount
FROM Movies m
JOIN Reservations res ON m.MovieId = res.MovieId
JOIN Seats s ON res.SeatId = s.SeatId
GROUP BY m.Genre
ORDER BY ReservationCount DESC;


--Q14 Calculate the total revenue generated by ticket sales for each movie
SELECT m.MovieTitle, SUM(p.Price) AS TotalRevenue
FROM Movies m
JOIN Prices p ON m.MovieId = p.MovieId
JOIN Reservations res ON m.MovieId = res.MovieId
GROUP BY m.MovieTitle;


--Q15 find the specific branch that is generating the most profits
SELECT TOP 1 b.BranchName, (SUM(p.Price) - SUM(f.Price)) AS Profit
FROM Branches b
JOIN Rooms r ON b.BranchID = r.BranchID
JOIN Movies m ON r.MovieId = m.MovieId
JOIN Prices p ON m.MovieId = p.MovieId
JOIN Reservations res ON m.MovieId = res.MovieId
JOIN Seats s ON res.SeatId = s.SeatId
JOIN Food f ON b.BranchID = f.BranchID
GROUP BY b.BranchName
ORDER BY Profit DESC;


--Q16 Retrieve the top 3 movies with the highest total revenue
SELECT TOP 3 m.MovieTitle, SUM(p.Price) AS TotalRevenue
FROM Movies m
```

```sql
JOIN Prices p ON m.MovieId = p.MovieId
JOIN Reservations r ON m.MovieId = r.MovieId
JOIN Seats s ON r.SeatId = s.SeatId
JOIN Rooms ro ON s.RoomId = ro.RoomId
JOIN Branches b ON ro.BranchID = b.BranchID
GROUP BY m.MovieTitle
ORDER BY TotalRevenue DESC


--Q17 Calculate the average duration of movies for each director:
SELECT m.Director, AVG(m.Duration) AS average_duration
FROM Movies m
JOIN Schedule s ON m.MovieId = s.MovieId
JOIN Rooms r ON s.RoomId = r.RoomId
GROUP BY m.Director;


-- Q18 Get the total number of reservations made by each customer:
SELECT c.CustomerName, COUNT(*) AS reservation_count
FROM LoyaltyCustomers c
JOIN Reservations r ON c.CustomerId = r.CustomerId
JOIN Movies m ON r.MovieId = m.MovieId
GROUP BY c.CustomerName;
```