Database Design Assignment
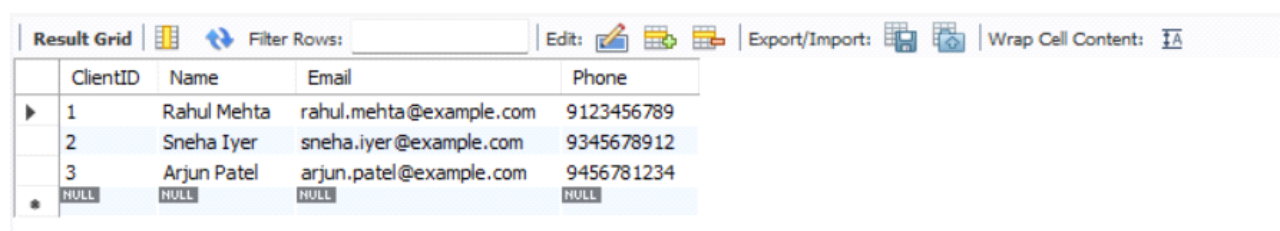
```
create database amds;
USE amds;

CREATE TABLE Clients (
    ClientID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Phone VARCHAR(15) UNIQUE NOT NULL
);

INSERT INTO Clients (Name, Email, Phone) VALUES
('Rahul Mehta', 'rahul.mehta@example.com', '9123456789'),
('Sneha Iyer', 'sneha.iyer@example.com', '9345678912'),
('Arjun Patel', 'arjun.patel@example.com', '9456781234');

SELECT * FROM Clients;
```

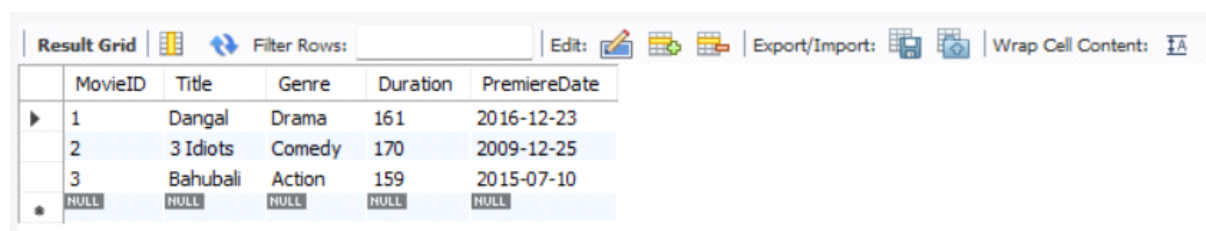|	ClientID	Name	Email	Phone
▶	1	Rahul Mehta	rahul.mehta@example.com	9123456789
	2	Sneha Iyer	sneha.iyer@example.com	9345678912
	3	Arjun Patel	arjun.patel@example.com	9456781234
*	NULL	NULL	NULL	NULL

```
CREATE TABLE Movies (
    MovieID INT AUTO_INCREMENT PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Genre VARCHAR(50),
    Duration INT CHECK (Duration > 0),
    PremiereDate DATE
);

INSERT INTO Movies (Title, Genre, Duration, PremiereDate) VALUES
('Dangal', 'Drama', 161, '2016-12-23'),
('3 Idiots', 'Comedy', 170, '2009-12-25'),
('Bahubali', 'Action', 159, '2015-07-10');

SELECT * FROM Movies;
```

	MovieID	Title	Genre	Duration	PremiereDate
▶	1	Dangal	Drama	161	2016-12-23
	2	3 Idiots	Comedy	170	2009-12-25
	3	Bahubali	Action	159	2015-07-10
*	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Theaters (
    TheaterID INT AUTO_INCREMENT PRIMARY KEY,
    Venue VARCHAR(100) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Capacity INT CHECK (Capacity > 0)
);

INSERT INTO Theaters (Venue, Address, Capacity) VALUES
('Cinepolis', 'MG Road, Bangalore', 350),
```

('PVR Icon', 'Phoenix Marketcity, Mumbai', 400);

SELECT * FROM Theaters;

TheaterID	Venue	Address	Capacity
1	Cinepolis	MG Road, Bangalore	350
2	PVR Icon	Phoenix Marketcity, Mumbai	400
NULL	NULL	NULL	NULL

```sql
CREATE TABLE Bookings (
    BookingID INT AUTO_INCREMENT PRIMARY KEY,
    ClientID INT,
    MovieID INT,
    TheaterID INT,
    BookingTimestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    SeatsBooked INT CHECK (SeatsBooked > 0),
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID) ON DELETE CASCADE,
    FOREIGN KEY (MovieID) REFERENCES Movies(MovieID) ON DELETE CASCADE,
    FOREIGN KEY (TheaterID) REFERENCES Theaters(TheaterID) ON DELETE CASCADE
);

INSERT INTO Bookings (ClientID, MovieID, TheaterID, SeatsBooked) VALUES
(1, 2, 1, 3),
(2, 3, 2, 2);

SELECT * FROM Bookings;
```

BookingID	ClientID	MovieID	TheaterID	BookingTimestamp	SeatsBooked
1	1	2	1	2025-03-14 13:52:32	3
2	2	3	2	2025-03-14 13:52:32	2
NULL	NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE ShowTimes (
    ScheduleID INT AUTO_INCREMENT PRIMARY KEY,
    MovieID INT,
    TheaterID INT,
    Date DATE NOT NULL,
    Time TIME NOT NULL,
    FOREIGN KEY (MovieID) REFERENCES Movies(MovieID) ON DELETE CASCADE,
    FOREIGN KEY (TheaterID) REFERENCES Theaters(TheaterID) ON DELETE CASCADE
);

INSERT INTO ShowTimes (MovieID, TheaterID, Date, Time) VALUES
(1, 1, '2025-01-15', '18:30:00'),
(2, 2, '2025-02-16', '20:00:00');

SELECT * FROM ShowTimes;
```

ScheduleID	MovieID	TheaterID	Date	Time
1	1	1	2025-01-15	18:30:00
2	2	2	2025-02-16	20:00:00
NULL	NULL	NULL	NULL	NULL

```sql
CREATE TABLE Reviews (
    ReviewID INT AUTO_INCREMENT PRIMARY KEY,
    ClientID INT,
    MovieID INT,
    Score INT CHECK (Score BETWEEN 1 AND 5),
```

Remarks TEXT,
        ReviewTimestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
        FOREIGN KEY (ClientID) REFERENCES Clients(ClientID) ON DELETE CASCADE,
        FOREIGN KEY (MovieID) REFERENCES Movies(MovieID) ON DELETE CASCADE
);

INSERT INTO Reviews (ClientID, MovieID, Score, Remarks) VALUES
(1, 2, 5, 'Excellent story and brilliant acting!'),
(2, 3, 4, 'Amazing action, a bit too long.');

SELECT * FROM Reviews;

	ReviewID	ClientID	MovieID	Score	Remarks	ReviewTimestamp
▶	1	1	2	5	Excellent story and brilliant acting!	2025-03-14 13:54:00
	2	2	3	4	Amazing action, a bit too long.	2025-03-14 13:54:00
*	NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Payments (
    PaymentID INT AUTO_INCREMENT PRIMARY KEY,
    ClientID INT,
    TotalPaid DECIMAL(10,2) NOT NULL,
    Method VARCHAR(50),
    State VARCHAR(20),
    PaymentTimestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID) ON DELETE CASCADE
);
```

INSERT INTO Payments (ClientID, TotalPaid, Method, State) VALUES
(1, 600.00, 'Debit Card', 'Completed'),
(2, 300.00, 'Net Banking', 'Pending');

SELECT * FROM Payments;

	PaymentID	ClientID	TotalPaid	Method	State	PaymentTimestamp
▶	1	1	600.00	Debit Card	Completed	2025-03-14 13:55:09
	2	2	300.00	Net Banking	Pending	2025-03-14 13:55:09
*	NULL	NULL	NULL	NULL	NULL	NULL

SELECT Clients.Name, Bookings.BookingID, Movies.Title, Bookings.SeatsBooked
FROM Bookings

	Name	BookingID	Title	SeatsBooked
▶	Rahul Mehta	1	3 Idiots	3
	Sneha Iyer	2	Bahubali	2

INNER JOIN Clients ON Bookings.ClientID = Clients.ClientID

	Name	BookingID	Title	SeatsBooked
▶	Rahul Mehta	1	3 Idiots	3
	Sneha Iyer	2	Bahubali	2

INNER JOIN Movies ON Bookings.MovieID = Movies.MovieID;

Name	BookingID	Title	SeatsBooked
Rahul Mehta	1	3 Idiots	3
Sneha Iyer	2	Bahubali	2

ER Diagram