Create the Database

CREATE DATABASE HotelManagement;

-- Use the Database

USE HotelManagement;

-- Create the Bill Table with Primary Key Constraint and Check Constraint

CREATE TABLE Bill (

billno INT PRIMARY KEY,

day DATE,

tableno INT CHECK (tableno > 0),

total DECIMAL(10, 2)

);

-- Create the Menu Table with Primary Key Constraint and Check Constraint

CREATE TABLE Menu (

dishno INT PRIMARY KEY,

dish\_desc VARCHAR(100),

price DECIMAL(10, 2) CHECK (price > 0)

);

SELECT SUM(total) AS Total\_Amount\_Collected

FROM Bill

WHERE day = '2013-08-01';

SELECT COUNT(m.dishno) AS NumberOfMenus

FROM Menu m

JOIN Bill b ON m.dishno = b.billno

WHERE b.billno = 2;

SELECT dishno, dish\_desc, price

FROM Menu

WHERE price BETWEEN 50 AND 200;

SELECT tableno, COUNT(billno) AS NumberOfBills

FROM Bill

WHERE day = '2013-12-01'

GROUP BY tableno;

SELECT b.billno, m.dish\_desc, m.price

FROM Bill b

INNER JOIN Menu m ON b.billno = m.dishno;

MONGODB

-use Hospital

-db.createCollection("Patients")

-db.Patients.insertMany([

{

patient\_id: 1,

name: "John Doe",

dob: "1985-06-15",

city: "Mumbai",

illness: "Cardiology",

doctor: "Dr. A",

treatment: "Medication",

admission\_date: new Date("2024-01-15"),

discharge\_date: new Date("2024-01-22")

},

{

patient\_id: 2,

name: "Jane Smith",

dob: "1990-08-20",

city: "Pune",

illness: "Orthopedics",

doctor: "Dr. B",

treatment: "Surgery",

admission\_date: new Date("2024-02-01"),

discharge\_date: new Date("2024-02-10")

},

{

patient\_id: 3,

name: "Robert Brown",

dob: "1975-03-10",

city: "Nagpur",

illness: "Neurology",

doctor: "Dr. C",

treatment: "Therapy",

admission\_date: new Date("2024-01-25"),

discharge\_date: new Date("2024-02-05")

},

{

patient\_id: 4,

name: "Emily Davis",

dob: "2000-11-05",

city: "Pune",

illness: "Cardiology",

doctor: "Dr. A",

treatment: "Medication",

admission\_date: new Date("2024-02-15"),

discharge\_date: new Date("2024-02-22")

},

{

patient\_id: 5,

name: "Michael Johnson",

dob: "1988-12-12",

city: "Mumbai",

illness: "Dermatology",

doctor: "Dr. D",

treatment: "Ointment",

admission\_date: new Date("2024-03-01"),

discharge\_date: new Date("2024-03-07")

}

]);

-db.Patients.find({ illness: "Cardiology" })

-db.Patients.find({

admission\_date: {

$gte: new Date("2024-01-15"),

$lte: new Date("2024-02-15")

}

})

-db.Patients.find({ illness: "Cardiology" }).sort({ admission\_date: 1 })

-db.Patients.updateOne(

{ patient\_id: 2 },

{ $set: { doctor: "Dr. E" } }

)

-db.Patients.aggregate([

{

$group: {

\_id: "$doctor",

count: { $sum: 1 }

}

}

])

Graph

Step 1: Create Nodes for Movies and Customers

CREATE (avengers:Movie {title: "Avengers", showtime: "2024-12-24 18:00", price: 150})

CREATE (inception:Movie {title: "Inception", showtime: "2024-12-24 20:00", price: 200})

CREATE (titanic:Movie {title: "Titanic", showtime: "2024-12-24 22:00", price: 100})

CREATE (mike:Customer {name: "Mike"})

CREATE (sarah:Customer {name: "Sarah"})

CREATE (john:Customer {name: "John"})

CREATE (lisa:Customer {name: "Lisa"})

CREATE (tia:Customer {name: "Tia"})

Create Relationships for Bookings

CREATE (mike)-[:BOOKED {seats: 3}]->(avengers)

CREATE (mike)-[:BOOKED {seats: 2}]->(inception)

CREATE (sarah)-[:BOOKED {seats: 4}]->(titanic)

CREATE (john)-[:BOOKED {seats: 2}]->(avengers)

MATCH (c:Customer)-[b:BOOKED]->(m:Movie {title: "Avengers"})

RETURN c.name, b.seats, m.title, m.showtime, m.price

MATCH (c:Customer)

WHERE NOT (c)-[:BOOKED]->()

RETURN c.name