**1. Hospital Management System**

**ER Diagram:**

plaintext

Copy code

+-----------+ +------------+ +-------------+

| Patient |<-------->| Appointment |<--------| Doctor |

+-----------+ +------------+ +-------------+

| PatientID | | AppointmentID | | DoctorID |

| Name | | PatientID | | Name |

| DOB | | DoctorID | | Specialty |

| Address | | AppointmentDate| +-------------+

+-----------+ +----------------+

|

v

+----------+

| Treatment|

+----------+

| TreatmentID|

| AppointmentID|

| Description |

+-------------+

**Database Schema:**

CREATE TABLE Patient (

PatientID INT PRIMARY KEY,

Name VARCHAR(100),

DOB DATE,

Address VARCHAR(255)

);

CREATE TABLE Doctor (

DoctorID INT PRIMARY KEY,

Name VARCHAR(100),

Specialty VARCHAR(100)

);

CREATE TABLE Appointment (

AppointmentID INT PRIMARY KEY,

PatientID INT,

DoctorID INT,

AppointmentDate DATE,

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),

FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)

);

CREATE TABLE Treatment (

TreatmentID INT PRIMARY KEY,

AppointmentID INT,

Description TEXT,

FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID)

);

**Queries:**

* **DML**: Insert a new patient.

sql

Copy code

INSERT INTO Patient (PatientID, Name, DOB, Address) VALUES (1, 'John Doe', '1990-01-01', '123 Elm Street');

* **DML**: Schedule an appointment.

sql

Copy code

INSERT INTO Appointment (AppointmentID, PatientID, DoctorID, AppointmentDate) VALUES (1, 1, 1, '2023-06-15');

* **DCL**: Grant select permission to a user.

sql

Copy code

GRANT SELECT ON Patient TO 'user';

* **Join**: Select all appointments with patient and doctor details.

sql

Copy code

SELECT a.AppointmentID, p.Name AS PatientName, d.Name AS DoctorName, a.AppointmentDate

FROM Appointment a

JOIN Patient p ON a.PatientID = p.PatientID

JOIN Doctor d ON a.DoctorID = d.DoctorID;

* **Union**: List all patients and doctors.

sql

Copy code

SELECT Name, 'Patient' AS Type FROM Patient

UNION

SELECT Name, 'Doctor' AS Type FROM Doctor;

* **Alter**: Add a new column to the Treatment table.

sql

Copy code

ALTER TABLE Treatment ADD COLUMN TreatmentDate DATE;

**2. Banking System**

**ER Diagram:**

plaintext

Copy code

+-----------+ +----------+ +-------------+

| Customer |<-------->| Account |<-------->| Branch |

+-----------+ +----------+ +-------------+

| CustomerID| | AccountID| | BranchID |

| Name | | CustomerID| | BranchName |

| Address | | BranchID | | BranchAddress|

| Phone | | Balance | +-------------+

+-----------+ +-----------+

|

v

+------------+

| Transaction|

+------------+

| TransactionID|

| AccountID |

| Date |

| Amount |

| Type |

+--------------+

**Database Schema:**

sql

Copy code

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY,

Name VARCHAR(100),

Address VARCHAR(255),

Phone VARCHAR(15)

);

CREATE TABLE Branch (

BranchID INT PRIMARY KEY,

BranchName VARCHAR(100),

BranchAddress VARCHAR(255)

);

CREATE TABLE Account (

AccountID INT PRIMARY KEY,

CustomerID INT,

BranchID INT,

Balance DECIMAL(10, 2),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID)

);

CREATE TABLE Transaction (

TransactionID INT PRIMARY KEY,

AccountID INT,

Date DATE,

Amount DECIMAL(10, 2),

Type VARCHAR(10),

FOREIGN KEY (AccountID) REFERENCES Account(AccountID)

);

**Queries:**

* **DML**: Insert a new account.

sql

Copy code

INSERT INTO Account (AccountID, CustomerID, BranchID, Balance) VALUES (1, 1, 1, 1000.00);

* **DML**: Record a transaction.

sql

Copy code

INSERT INTO Transaction (TransactionID, AccountID, Date, Amount, Type) VALUES (1, 1, '2023-06-05', 200.00, 'Deposit');

* **DCL**: Revoke insert permission from a user.

sql

Copy code

REVOKE INSERT ON Account FROM 'user';

* **Join**: Select all transactions with customer details.

sql

Copy code

SELECT t.TransactionID, c.Name AS CustomerName, t.Date, t.Amount, t.Type

FROM Transaction t

JOIN Account a ON t.AccountID = a.AccountID

JOIN Customer c ON a.CustomerID = c.CustomerID;

* **Union**: List all customers and branch names.

sql

Copy code

SELECT Name, 'Customer' AS Type FROM Customer

UNION

SELECT BranchName, 'Branch' AS Type FROM Branch;

* **Alter**: Add a new column to the Account table.

sql

Copy code

ALTER TABLE Account ADD COLUMN AccountType VARCHAR(50);

**3. Airline Reservation System**

**ER Diagram:**

plaintext

Copy code

+-----------+ +---------+ +----------+

| Passenger |<-------->| Booking |<-------->| Flight |

+-----------+ +---------+ +----------+

| PassengerID| | BookingID| | FlightID |

| Name | | PassengerID| | FlightNum|

| PassportNo| | FlightID | | Departure|

+-----------+ | BookingDate| | Arrival |

+------------+ +----------+

|

v

+-------+

| Seat |

+-------+

| SeatID|

| FlightID|

| SeatNum|

| Class |

+--------+

**Database Schema:**

sql

Copy code

CREATE TABLE Passenger (

PassengerID INT PRIMARY KEY,

Name VARCHAR(100),

PassportNumber VARCHAR(20)

);

CREATE TABLE Flight (

FlightID INT PRIMARY KEY,

FlightNumber VARCHAR(10),

Departure VARCHAR(100),

Arrival VARCHAR(100)

);

CREATE TABLE Booking (

BookingID INT PRIMARY KEY,

PassengerID INT,

FlightID INT,

BookingDate DATE,

FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID),

FOREIGN KEY (FlightID) REFERENCES Flight(FlightID)

);

CREATE TABLE Seat (

SeatID INT PRIMARY KEY,

FlightID INT,

SeatNumber VARCHAR(5),

Class VARCHAR(20),

FOREIGN KEY (FlightID) REFERENCES Flight(FlightID)

);

**Queries:**

* **DML**: Book a seat.

sql

Copy code

INSERT INTO Booking (BookingID, PassengerID, FlightID, BookingDate) VALUES (1, 1, 1, '2023-06-10');

* **DML**: Assign a seat.

sql

Copy code

INSERT INTO Seat (SeatID, FlightID, SeatNumber, Class) VALUES (1, 1, '12A', 'Economy');

* **DCL**: Grant update permission to a user.

sql

Copy code

GRANT UPDATE ON Seat TO 'user';

* **Join**: Select all bookings with passenger and flight details.

sql

Copy code

SELECT b.BookingID, p.Name AS PassengerName, f.FlightNumber, b.BookingDate

FROM Booking b

JOIN Passenger p ON b.PassengerID = p.PassengerID

JOIN Flight f ON b.FlightID = f.FlightID;

* **Union**: List all passengers and flights.

sql

Copy code

SELECT Name, 'Passenger' AS Type FROM Passenger

UNION

SELECT FlightNumber, 'Flight' AS Type FROM Flight;

* **Alter**: Add a new column to the Flight table.

sql

Copy code

ALTER TABLE Flight ADD COLUMN Airline VARCHAR(100);

**4. Payroll System**

**ER Diagram:**

plaintext

Copy code

+-----------+ +-----------+ +--------+

| Employee |<-------->| Department|<-------->| Salary |

+-----------+ +-----------+ +--------+

| EmployeeID| | DepartmentID| | SalaryID|

| Name | | DeptName | | EmployeeID|

| DeptID | +-----------+ | Amount |

| Position | | Date |

+-----------+ +---------+

|

v

+---------+

| Payroll |

+---------+

| PayrollID|

| EmployeeID|

| SalaryID |

| PayDate |

+----------+

**Database Schema:**

sql

Copy code

CREATE TABLE Employee (

EmployeeID INT PRIMARY KEY,

Name VARCHAR(100),

DepartmentID INT,

Position VARCHAR(100),

FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)

);

CREATE TABLE Department (

DepartmentID INT PRIMARY KEY,

DeptName VARCHAR(100)

);

CREATE TABLE Salary (

SalaryID INT PRIMARY KEY,

EmployeeID INT,

Amount DECIMAL(10, 2),

Date DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

);

CREATE TABLE Payroll (

PayrollID INT PRIMARY KEY,

EmployeeID INT,

SalaryID INT,

PayDate DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),

FOREIGN KEY (SalaryID) REFERENCES Salary(SalaryID)

);

**Queries:**

* **DML**: Insert a new employee.

sql

Copy code

INSERT INTO Employee (EmployeeID, Name, DepartmentID, Position) VALUES (1, 'Alice', 1, 'Manager');

* **DML**: Record a salary.

sql

Copy code

INSERT INTO Salary (SalaryID, EmployeeID, Amount, Date) VALUES (1, 1, 5000.00, '2023-06-01');

* **DCL**: Grant select permission to a user.

sql

Copy code

GRANT SELECT ON Employee TO 'user';

* **Join**: Select all payroll records with employee and salary details.

sql

Copy code

SELECT p.PayrollID, e.Name AS EmployeeName, s.Amount, p.PayDate

FROM Payroll p

JOIN Employee e ON p.EmployeeID = e.EmployeeID

JOIN Salary s ON p.SalaryID = s.SalaryID;

* **Union**: List all employees and departments.

sql

Copy code

SELECT Name, 'Employee' AS Type FROM Employee

UNION

SELECT DeptName, 'Department' AS Type FROM Department;

* **Alter**: Add a new column to the Salary table.

sql

Copy code

ALTER TABLE Salary ADD COLUMN Bonus DECIMAL(10, 2);

**5. University Subject Allocation System**

**ER Diagram:**

plaintext

Copy code

+---------+ +---------+ +---------+

| Student |<-------->| Enroll |<-------->| Subject |

+---------+ +---------+ +---------+

| StudentID| | EnrollID| | SubjectID|

| Name | | StudentID| | Name |

| Major | | SubjectID| | Credits |

+---------+ | Semester | +---------+

+---------+

|

v

+--------+

| Teacher |

+--------+

| TeacherID|

| Name |

| SubjectID|

+---------+

**Database Schema:**

sql

Copy code

CREATE TABLE Student (

StudentID INT PRIMARY KEY,

Name VARCHAR(100),

Major VARCHAR(100)

);

CREATE TABLE Subject (

SubjectID INT PRIMARY KEY,

Name VARCHAR(100),

Credits INT

);

CREATE TABLE Enroll (

EnrollID INT PRIMARY KEY,

StudentID INT,

SubjectID INT,

Semester VARCHAR(10),

FOREIGN KEY (StudentID) REFERENCES Student(StudentID),

FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)

);

CREATE TABLE Teacher (

TeacherID INT PRIMARY KEY,

Name VARCHAR(100),

SubjectID INT,

FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)

);

**Queries:**

* **DML**: Enroll a student in a subject.

sql

Copy code

INSERT INTO Enroll (EnrollID, StudentID, SubjectID, Semester) VALUES (1, 1, 1, 'Fall2023');

* **DML**: Assign a teacher to a subject.

sql

Copy code

INSERT INTO Teacher (TeacherID, Name, SubjectID) VALUES (1, 'Dr. Smith', 1);

* **DCL**: Grant select permission to a user.

sql

Copy code

GRANT SELECT ON Student TO 'user';

* **Join**: Select all enrollments with student and subject details.

sql

Copy code

SELECT e.EnrollID, s.Name AS StudentName, sub.Name AS SubjectName, e.Semester

FROM Enroll e

JOIN Student s ON e.StudentID = s.StudentID

JOIN Subject sub ON e.SubjectID = sub.SubjectID;

* **Union**: List all students and teachers.

sql

Copy code

SELECT Name, 'Student' AS Type FROM Student

UNION

SELECT Name, 'Teacher' AS Type FROM Teacher;

* **Alter**: Add a new column to the Subject table.

sql

Copy code

ALTER TABLE Subject ADD COLUMN Description TEXT;

**HR Recruitment System**

**ER Diagram:**

plaintext

Copy code

+----------------+ +-----------------+ +----------------+

| Candidate | | Application | | JobPost |

+----------------+ +-----------------+ +----------------+

| CandidateID |<---| ApplicationID |--->| JobID |

| Name | | CandidateID | | Title |

| Email | | JobID | | Description |

| Phone | | ApplicationDate | | Requirements |

+----------------+ +-----------------+ +----------------+

|

v

+----------+

| Interview|

+----------+

| InterviewID|

| ApplicationID|

| Date |

| Status |

+-------------+

**Database Schema:**

sql

Copy code

CREATE TABLE Candidate (

CandidateID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(15)

);

CREATE TABLE JobPost (

JobID INT PRIMARY KEY,

Title VARCHAR(100),

Description TEXT,

Requirements TEXT

);

CREATE TABLE Application (

ApplicationID INT PRIMARY KEY,

CandidateID INT,

JobID INT,

ApplicationDate DATE,

FOREIGN KEY (CandidateID) REFERENCES Candidate(CandidateID),

FOREIGN KEY (JobID) REFERENCES JobPost(JobID)

);

CREATE TABLE Interview (

InterviewID INT PRIMARY KEY,

ApplicationID INT,

Date DATE,

Status VARCHAR(50),

FOREIGN KEY (ApplicationID) REFERENCES Application(ApplicationID)

);

**Queries:**

* **DML**: Insert a new candidate.

sql

Copy code

INSERT INTO Candidate (CandidateID, Name, Email, Phone) VALUES (1, 'John Doe', 'john.doe@example.com', '1234567890');

* **DML**: Candidate applies for a job.

sql

Copy code

INSERT INTO Application (ApplicationID, CandidateID, JobID, ApplicationDate) VALUES (1, 1, 1, '2024-06-05');

* **DML**: Schedule an interview.

sql

Copy code

INSERT INTO Interview (InterviewID, ApplicationID, Date, Status) VALUES (1, 1, '2024-06-10', 'Scheduled');

* **DCL**: Grant select permission to a user.

sql

Copy code

GRANT SELECT ON Candidate TO 'recruiter';

* **Join**: Select all applications with candidate and job post details.

sql

Copy code

SELECT a.ApplicationID, c.Name AS CandidateName, j.Title AS JobTitle, a.ApplicationDate

FROM Application a

JOIN Candidate c ON a.CandidateID = c.CandidateID

JOIN JobPost j ON a.JobID = j.JobID;

* **Union**: List all candidates and job titles.

sql

Copy code

SELECT Name, 'Candidate' AS Type FROM Candidate

UNION

SELECT Title, 'JobPost' AS Type FROM JobPost;

* **Alter**: Add a new column to the Interview table.

sql

Copy code

ALTER TABLE Interview ADD COLUMN Feedback TEXT;

**Departmental Store Maintenance**

**ER Diagram:**

plaintext

Copy code

+-------------+ +------------+ +-------------+

| Product | | Inventory | | Supplier |

+-------------+ +------------+ +-------------+

| ProductID |<---| InventoryID|--->| SupplierID |

| Name | | ProductID | | Name |

| Description | | SupplierID | | ContactInfo |

| Price | | Quantity | | Address |

+-------------+ +------------+ +-------------+

|

v

+-----------+

| Order |

+-----------+

| OrderID |

| ProductID |

| SupplierID|

| OrderDate |

| Quantity |

+-----------+

**Database Schema:**

sql

Copy code

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

Name VARCHAR(100),

Description TEXT,

Price DECIMAL(10, 2)

);

CREATE TABLE Supplier (

SupplierID INT PRIMARY KEY,

Name VARCHAR(100),

ContactInfo VARCHAR(100),

Address VARCHAR(255)

);

CREATE TABLE Inventory (

InventoryID INT PRIMARY KEY,

ProductID INT,

SupplierID INT,

Quantity INT,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)

);

CREATE TABLE Order (

OrderID INT PRIMARY KEY,

ProductID INT,

SupplierID INT,

OrderDate DATE,

Quantity INT,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)

);

**Queries:**

* **DML**: Insert a new product.

sql

Copy code

INSERT INTO Product (ProductID, Name, Description, Price) VALUES (1, 'Laptop', 'Gaming Laptop', 1500.00);

* **DML**: Add a new supplier.

sql

Copy code

INSERT INTO Supplier (SupplierID, Name, ContactInfo, Address) VALUES (1, 'Tech Supplies Inc.', '123-456-7890', '456 Tech Avenue');

* **DML**: Update inventory.

sql

Copy code

INSERT INTO Inventory (InventoryID, ProductID, SupplierID, Quantity) VALUES (1, 1, 1, 50);

* **DML**: Create an order.

sql

Copy code

INSERT INTO Order (OrderID, ProductID, SupplierID, OrderDate, Quantity) VALUES (1, 1, 1, '2024-06-05', 10);

* **DCL**: Grant update permission to a user.

sql

Copy code

GRANT UPDATE ON Inventory TO 'store\_manager';

* **Join**: Select all orders with product and supplier details.

sql

Copy code

SELECT o.OrderID, p.Name AS ProductName, s.Name AS SupplierName, o.OrderDate, o.Quantity

FROM Order o

JOIN Product p ON o.ProductID = p.ProductID

JOIN Supplier s ON o.SupplierID = s.SupplierID;

* **Union**: List all products and suppliers.

sql

Copy code

SELECT Name, 'Product' AS Type FROM Product

UNION

SELECT Name, 'Supplier' AS Type FROM Supplier;

* **Alter**: Add a new column to the Product table.

sql

Copy code

ALTER TABLE Product ADD COLUMN StockLevel INT;

**Sport or Athletics Event Conductance System**

**ER Diagram:**

plaintext

Copy code

+-----------+ +----------+ +-------------+

| Event | | Athlete | | Sponsor |

+-----------+ +----------+ +-------------+

| EventID |<---| AthleteID|--->| SponsorID |

| Name | | Name | | Name |

| Date | | Age | | ContactInfo |

| Location | | Gender | | Address |

+-----------+ +----------+ +-------------+

|

v

+-----------+

| Result |

+-----------+

| ResultID |

| EventID |

| AthleteID |

| Position |

+-----------+

**Database Schema:**

sql

Copy code

CREATE TABLE Event (

EventID INT PRIMARY KEY,

Name VARCHAR(100),

Date DATE,

Location VARCHAR(100)

);

CREATE TABLE Athlete (

AthleteID INT PRIMARY KEY,

Name VARCHAR(100),

Age INT,

Gender VARCHAR(10)

);

CREATE TABLE Sponsor (

SponsorID INT PRIMARY KEY,

Name VARCHAR(100),

ContactInfo VARCHAR(100),

Address VARCHAR(255)

);

CREATE TABLE Result (

ResultID INT PRIMARY KEY,

EventID INT,

AthleteID INT,

Position INT,

FOREIGN KEY (EventID) REFERENCES Event(EventID),

FOREIGN KEY (AthleteID) REFERENCES Athlete(AthleteID)

);

**Queries:**

* **DML**: Insert a new event.

sql

Copy code

INSERT INTO Event (EventID, Name, Date, Location) VALUES (1, 'Marathon', '2024-06-20', 'Central Park');

* **DML**: Add a new athlete.

sql

Copy code

INSERT INTO Athlete (AthleteID, Name, Age, Gender) VALUES (1, 'Jane Smith', 25, 'Female');

* **DML**: Record a result.

sql

Copy code

INSERT INTO Result (ResultID, EventID, AthleteID, Position) VALUES (1, 1, 1, 1);

* **DML**: Add a new sponsor.

sql

Copy code

INSERT INTO Sponsor (SponsorID, Name, ContactInfo, Address) VALUES (1, 'Healthy Life Co.', 'contact@healthylife.com', '789 Wellness Way');

* **DCL**: Grant insert permission to a user.

sql

Copy code

GRANT INSERT ON Result TO 'event\_manager';

* **Join**: Select all results with event and athlete details.

sql

Copy code

SELECT r.ResultID, e.Name AS EventName, a.Name AS AthleteName, r.Position

FROM Result r

JOIN Event e ON r.EventID = e.EventID

JOIN Athlete a ON r.AthleteID = a.AthleteID;

* **Union**: List all events and sponsors.

sql

Copy code

SELECT Name, 'Event' AS Type FROM Event

UNION

SELECT Name, 'Sponsor' AS Type FROM Sponsor;

* **Alter**: Add a new column to the Athlete table.

sql

Copy code

ALTER TABLE Athlete ADD COLUMN Nationality VARCHAR(50);

4o