



Department of Artificial Intelligence and Machine Learning

23CST201 DATABASE MANAGEMENT SYSTEMS

Assignment I – CO1

Construct the E R Diagram for the following scenario

1. Supermarket Management System

A supermarket wants to manage its inventory efficiently. The system should track products, their suppliers, and customers. Each product has a unique ID, name, category, price, and stock level. Suppliers provide multiple products and have supplier IDs, names, and contact details. Customers make purchases and have customer IDs, names, and purchase history.

Entities & Attributes

- Product (ProductID, Name, Category, Price, StockLevel)
- Supplier (SupplierID, Name, Contact, Address)
- Customer (CustomerID, Name, Contact, PurchaseHistory)
- Order (OrderID, OrderDate, TotalAmount)

Relationships

- Supplier supplies Product (1:M)
- Customer places Order (1:M)
- Order contains Product (M:N)

2. Restaurant Management System

A restaurant requires a system to manage customers, orders, and menu items. Each customer can place multiple orders. The system should store menu items with their prices and ingredients, and track chefs preparing meals.

Entities & Attributes

- Customer (*CustomerID, Name, Contact*)
- Order (*OrderID, OrderDate, TotalPrice*)
- MenuItem (*ItemID, Name, Price, Category*)
- Chef (*ChefID, Name, Specialization*)

Relationships

- Customer places Order (1:M)
- Order contains MenuItem (M:N)
- Chef prepares MenuItem (M:N)

3. Shopping Mall Management System

A shopping mall consists of multiple stores, each having an owner, employees, and customers. The mall administration wants to track store leases, sales, and maintenance schedules.

Entities & Attributes

- Mall (*MallID, Name, Location*)
- Store (*StoreID, Name, Type, OwnerID*)
- Owner (*OwnerID, Name, Contact*)
- Customer (*CustomerID, Name, Email*)

Relationships

- Mall contains Store (1:M)
- Store has Owner (1:1)

- Customer visits Store ($M:N$)

4. Car Rental System

A car rental service rents cars to customers. The system should store customer details, car details, booking history, rental rates, and return status.

Entities & Attributes

- Car (*CarID, Model, Make, RentalPrice*)
- Customer (*CustomerID, Name, LicenseNumber, Contact*)
- Rental (*RentalID, StartDate, EndDate, TotalCost*)

Relationships

- Customer rents Car ($M:N$)
- Rental involves Car ($M:1$)

5. E-Commerce Platform

An e-commerce website sells products from multiple vendors to customers. It should track orders, reviews, payments, delivery status, and product availability.

Entities & Attributes

- Product (*ProductID, Name, Price, Stock*)
- Vendor (*VendorID, Name, Contact*)
- Customer (*CustomerID, Name, Address, Email*)
- Order (*OrderID, Date, TotalPrice*)

Relationships

- Vendor supplies Product ($1:M$)
- Customer places Order ($1:M$)
- Order contains Product ($M:N$)

6. Real Estate Agency

A real estate agency sells and rents properties. The system must store property details, agent details, buyer information, and transactions.

Entities & Attributes:

- Property (*PropertyID, Address, Type, Price, Status*)
- Agent (*AgentID, Name, Contact, CommissionRate*)
- Buyer (*BuyerID, Name, Contact, Budget*)
- Seller (*SellerID, Name, Contact, PropertyID*)

Relationships:

- Seller owns Property ($1:M$)
- Agent lists Property ($1:M$)
- Buyer purchases Property ($M:N$)
- Agent assists Buyer ($M:N$)

7. Online Ticket Booking System

A system to book bus, train, and flight tickets. It should track passenger details, available seats, ticket prices, and payment methods.

Entities & Attributes:

- User (*UserID, Name, Email, Contact, PaymentInfo*)
- Event (*EventID, Name, Date, Venue, Price*)
- Booking (*BookingID, BookingDate, TotalAmount, Status*)
- Payment (*PaymentID, PaymentMethod, PaymentStatus*)

Relationships:

- User books Event ($M:N$)
- Booking is associated with User ($M:1$)
- Booking includes Payment ($1:1$)

8. Hotel Management System

A hotel chain wants to track room reservations, services, and guests. The system should store check-in/check-out details, room types, and staff assignments.

Entities & Attributes:

- Hotel (*HotelID, Name, Location, Rating*)
- Room (*RoomID, Type, Price, Availability, HotelID*)

- Guest (*GuestID, Name, Contact, Email*)
- Booking (*BookingID, CheckInDate, CheckOutDate, TotalAmount, Status*)
- Payment (*PaymentID, PaymentMethod, PaymentStatus, BookingID*)

Relationships:

- Hotel has Room (*1:M*)
- Guest makes Booking (*1:M*)
- Booking includes Room (*M:N*)
- Booking is associated with Payment (*1:1*)

9. Library Management System

A library needs to track books, members, borrow history, and late fees. Each book has a unique ID, title, author, and category.

Entities & Attributes:

- **Book** (*BookID, Title, Author, ISBN, Genre, Availability*)
- **Member** (*MemberID, Name, Contact, Email, MembershipType*)
- **Librarian** (*LibrarianID, Name, Contact, Shift*)
- **Loan** (*LoanID, IssueDate, DueDate, ReturnDate, Fine*)

Relationships:

- Member **borrow**s Book (*M:N*)
- Librarian **manages** Book (*1:M*)
- Loan **is associated with** Book and Member (*M:N*)

10. Delivery Service Management

A delivery service wants to manage orders, delivery persons, and payment status. The system should track real-time delivery status.

Entities & Attributes:

- **Customer** (*CustomerID, Name, Address, Contact*)
- **Order** (*OrderID, OrderDate, DeliveryAddress, Status*)
- **DeliveryAgent** (*AgentID, Name, Contact, VehicleType*)
- **Package** (*PackageID, Weight, Dimensions, Type, Status*)

Relationships:

- Customer **places** Order (*1:M*)
- Order **contains** Package (*1:M*)
- DeliveryAgent **delivers** Order (*1:M*)
- Package **is assigned to** DeliveryAgent (*M:1*)

11. University Management System

A university needs to track students, courses, professors, and departments. Each student enrolls in multiple courses and has a unique student ID.

Entities & Attributes:

- **Student** (*StudentID, Name, Age, Major*)
- **Course** (*CourseID, Name, Credits*)
- **Professor** (*ProfessorID, Name, Department*)

Relationships:

- Student **enrolls in** Course (*M:N*)
- Professor **teaches** Course (*1:M*)

12. School Management System

A school wants to manage students, classes, teachers, and grades. The system should include attendance, timetable, and exams.

Entities & Attributes:

- **Student** (*StudentID, Name, Grade*)
- **Teacher** (*TeacherID, Name, Subject*)
- **Class** (*ClassID, Name, Section*)

Relationships:

- Teacher **teaches** Class (*1:M*)
- Student **attends** Class (*M:N*)

13. College Admission System

A college admission system tracks applications, student eligibility, fees, and seat allocation.

Entities & Attributes:

- Student (StudentID, Name, Age, Qualification, EntranceExamScore, ContactNumber, Email, Address)
- Course (CourseID, Name, Duration, Department, EligibilityCriteria, TotalSeats, AvailableSeats)
- Application (ApplicationID, StudentID, CourseID, ApplicationDate, Status, MeritScore)
- FeePayment (PaymentID, StudentID, CourseID, PaymentDate, Amount, PaymentStatus, TransactionID)
- SeatAllocation (AllocationID, StudentID, CourseID, AllocationStatus, SeatNumber)
- Department (DepartmentID, Name, HOD)
- AdmissionOfficer (OfficerID, Name, Email, ContactNumber, Role)

Relationships:

- Student applies for Course (*M:N*)
- Student makes FeePayment (*1:1*)
- Student gets SeatAllocation (*1:1*)
- Course belongs to Department (*M:1*)
- AdmissionOfficer reviews Application (*1:M*)

14. Online Learning Platform

An e-learning platform offers courses, quizzes, certificates, and teacher-student interactions.

Entities & Attributes:

- **Student** (StudentID, Name, Age, Email, ContactNumber, EnrollmentDate)
- **Course** (CourseID, Name, Category, Duration, DifficultyLevel, Price)
- **Instructor** (InstructorID, Name, Email, Expertise, Experience, ContactNumber)
- **Enrollment** (EnrollmentID, StudentID, CourseID, EnrollmentDate, Status)
- **Quiz** (QuizID, CourseID, Title, TotalQuestions, PassingScore)
- **QuizAttempt** (AttemptID, StudentID, QuizID, Score, AttemptDate, Status)
- **Certificate** (CertificateID, StudentID, CourseID, IssueDate, Validity)
- **Interaction** (InteractionID, StudentID, InstructorID, Message, Timestamp, InteractionType)

Relationships:

- **Student enrolls in Course** (*M:N*)
- **Instructor teaches Course** (*1:M*)
- **Student attempts Quiz** (*M:N*)
- **Student earns Certificate for Course** (*1:1*)
- **Student interacts with Instructor** (*M:N*)

15. Research Organization

A research center tracks projects, researchers, publications, and funding.

Entities & Attributes:

- **Researcher** (ResearcherID, Name, Email, Specialization, Experience, ContactNumber)
- **Project** (ProjectID, Title, StartDate, EndDate, Status, FundingAmount, ResearcherID)
- **Publication** (PublicationID, Title, Journal, PublicationDate, DOI, ResearcherID, ProjectID)
- **Funding** (FundingID, SponsorName, Amount, FundingDate, ProjectID)
- **Department** (DepartmentID, Name, Head, ContactNumber)

Relationships:

- **Researcher works on Project** (*M:N*)
- **Project receives Funding** (*1:M*)
- **Researcher publishes Publication** (*1:M*)
- **Publication is related to Project** (*M:1*)
- **Researcher belongs to Department** (*M:1*)

16. Internship Portal

A system to connect students and companies offering internships, including applications, eligibility, and feedback.

Entities & Attributes:

- **Student** (StudentID, Name, Age, Email, ContactNumber, Qualification, Skills, Resume)
- **Company** (CompanyID, Name, Industry, Location, ContactPerson, Email, ContactNumber)
- **Internship** (InternshipID, Title, CompanyID, Duration, Stipend, EligibilityCriteria, ApplicationDeadline)
- **Application** (ApplicationID, StudentID, InternshipID, ApplicationDate, Status)
- **Feedback** (FeedbackID, InternshipID, StudentID, CompanyID, Rating, Comments, FeedbackDate)

Relationships:

- **Student applies for Internship** (*M:N*)
- **Company offers Internship** (*1:M*)
- **Student receives Feedback from Company** (*1:1*)
- **Company provides Feedback on Student** (*1:1*)

17. Coaching Institute

A coaching center tracks students, batches, courses, and test scores.

Entities & Attributes:

- **Student** (StudentID, Name, Age, Email, ContactNumber, Address)
- **Course** (CourseID, Name, Duration, Fees)
- **Batch** (BatchID, CourseID, BatchStartDate, BatchEndDate, InstructorID, Timing)
- **Instructor** (InstructorID, Name, Email, Specialization, ContactNumber)
- **Test** (TestID, CourseID, Date, TotalMarks)
- **Score** (ScoreID, StudentID, TestID, MarksObtained, Percentage, Result)

Relationships:

- **Student enrolls in Batch** (*M:N*)
- **Batch belongs to Course** (*M:1*)
- **Instructor teaches Batch** (*1:M*)
- **Student takes Test** (*M:N*)
- **Test is part of Course** (*M:1*)
- **Student receives Score for Test** (*1:1*)

18. Student Hostel Management

A hostel needs to track room assignments, students, and staff.

Entities & Attributes:

- **Student** (StudentID, Name, Age, Email, ContactNumber, RoomID, CheckInDate, CheckOutDate)
- **Room** (RoomID, RoomNumber, Type, Capacity, Status)
- **Staff** (StaffID, Name, Role, ContactNumber, ShiftTimings)
- **MaintenanceRequest** (RequestID, RoomID, StudentID, RequestDate, Status, Description)
- **FeePayment** (PaymentID, StudentID, Amount, DueDate, PaymentStatus)

Relationships:

- **Student is assigned to Room** (*1:1*)
- **Room is maintained by Staff** (*M:N*)
- **Student raises MaintenanceRequest** (*M:N*)
- **Student makes FeePayment** (*1:1*)

19. Conference Management System

A conference tracks attendees, sessions, and guest speakers.

Entities & Attributes:

- **Attendee** (AttendeeID, Name, Email, ContactNumber, Organization, RegistrationStatus)
- **Session** (SessionID, Title, SpeakerID, Date, Time, Duration, RoomNumber)
- **Speaker** (SpeakerID, Name, Email, Topic, Affiliation)
- **Conference** (ConferenceID, Name, Location, StartDate, EndDate, Organizer)
- **Registration** (RegistrationID, AttendeeID, ConferenceID, RegistrationDate, PaymentStatus)

Relationships:

- **Attendee registers for Conference** ($M:N$)
- **Conference has Sessions** ($1:M$)
- **Speaker delivers Session** ($1:M$)
- **Attendee attends Session** ($M:N$)

20. Examination System

A system to manage exams, question papers, student scores, and result generation.

Entities & Attributes:

- **Student** (StudentID, Name, Age, Email, ContactNumber, CourseID)
- **Exam** (ExamID, CourseID, Date, Duration, TotalMarks)
- **QuestionPaper** (PaperID, ExamID, NumberOfQuestions, DifficultyLevel)
- **Question** (QuestionID, PaperID, QuestionText, Marks)
- **AnswerSheet** (SheetID, StudentID, ExamID, TotalScore, Result)

Relationships:

- **Student takes Exam** ($M:N$)
- **Exam has QuestionPaper** ($1:1$)
- **QuestionPaper contains Questions** ($1:M$)
- **Student submits AnswerSheet** ($1:1$)

21. Hospital Management System

A hospital tracks patients, doctors, appointments, and medical records.

Entities & Attributes:

- **Patient** (PatientID, Name, Age, Gender, ContactNumber, Address, MedicalHistory)
- **Doctor** (DoctorID, Name, Specialization, ContactNumber, Email)
- **Appointment** (AppointmentID, PatientID, DoctorID, Date, Time, Status)
- **MedicalRecord** (RecordID, PatientID, Diagnosis, Treatment, Prescription, Date)

Relationships:

- **Patient books Appointment with Doctor** ($M:N$)
- **Doctor maintains MedicalRecord for Patient** ($1:M$)

22. Blood Bank Management System

A blood bank maintains donor details, blood groups, and stock levels.

Entities & Attributes:

- **Donor** (DonorID, Name, Age, BloodGroup, ContactNumber, LastDonationDate)
- **BloodStock** (StockID, BloodGroup, QuantityAvailable, ExpiryDate)
- **Donation** (DonationID, DonorID, Date, VolumeDonated)
- **Request** (RequestID, PatientName, BloodGroup, Hospital, RequestDate, Status)

Relationships:

- **Donor makes a Donation** ($1:M$)
- **Hospital requests BloodStock** ($M:N$)

23. Pharmacy System

A pharmacy tracks medicines, suppliers, and prescriptions.

Entities & Attributes:

- **Medicine** (MedicineID, Name, Type, ExpiryDate, Price, StockQuantity)
- **Supplier** (SupplierID, Name, ContactNumber, Address, Email)
- **Prescription** (PrescriptionID, PatientID, DoctorID, MedicineID, Dosage, Duration)
- **Purchase** (PurchaseID, CustomerID, MedicineID, Quantity, TotalPrice, PurchaseDate)

Relationships:

- **Pharmacy purchases Medicine from Supplier** (*M:N*)
- **Doctor prescribes Medicine for Patient** (*M:N*)

24. Telemedicine Platform

An online consultation platform connects patients with doctors via video calls.

Entities & Attributes:

- **Patient** (PatientID, Name, Age, ContactNumber, Email)
- **Doctor** (DoctorID, Name, Specialization, ContactNumber, Email)
- **Consultation** (ConsultationID, PatientID, DoctorID, Date, Time, Duration, Diagnosis)
- **Prescription** (PrescriptionID, ConsultationID, MedicineName, Dosage, Duration)

Relationships:

- **Patient consults Doctor online** (*M:N*)
- **Doctor provides Prescription for Patient** (*1:M*)

25. Health Insurance Company

A health insurance firm tracks policyholders, claims, and hospitals.

Entities & Attributes:

- **Policyholder** (PolicyholderID, Name, Age, ContactNumber, PolicyType, PremiumAmount)
- **Policy** (PolicyID, PolicyholderID, StartDate, EndDate, CoverageAmount)
- **Claim** (ClaimID, PolicyholderID, HospitalID, AmountClaimed, ClaimStatus)
- **Hospital** (HospitalID, Name, Location, ContactNumber)

Relationships:

- **Policyholder holds Policy** (*1:M*)
- **Policyholder makes Claim at Hospital** (*M:N*)

26. Ambulance Service

A system for tracking emergency requests, locations, and hospital availability.

Entities & Attributes:

- **Ambulance** (AmbulanceID, VehicleNumber, DriverName, ContactNumber, Status)
- **Request** (RequestID, PatientName, Location, ContactNumber, EmergencyType, Status)
- **Hospital** (HospitalID, Name, Address, ContactNumber, BedAvailability)
- **Dispatch** (DispatchID, AmbulanceID, RequestID, HospitalID, DispatchTime, ArrivalTime)

Relationships:

- **Patient requests Ambulance** (*M:N*)
- **Ambulance transports Patient to Hospital** (*M:N*)

27. Veterinary Clinic

A pet hospital tracks owners, pets, and medical records.

Entities & Attributes:

- **Owner** (OwnerID, Name, ContactNumber, Address, Email)
- **Pet** (PetID, OwnerID, Name, Species, Breed, Age, MedicalHistory)
- **Veterinarian** (VetID, Name, Specialization, ContactNumber)
- **Appointment** (AppointmentID, PetID, VetID, Date, Time, Diagnosis)

Relationships:

- **Owner has Pet** (*1:M*)
- **Pet has Appointment with Veterinarian** (*M:N*)

28. Organ Donor Registry

A system to manage donors and recipients for organ transplants.

Entities & Attributes:

- **Donor** (DonorID, Name, Age, BloodGroup, ContactNumber, OrganType)
- **Recipient** (RecipientID, Name, Age, BloodGroup, ContactNumber, RequiredOrgan)
- **Hospital** (HospitalID, Name, Location, ContactNumber)

- **Transplant** (TransplantID, DonorID, RecipientID, HospitalID, Date, Status)

Relationships:

- **Donor donates Organ to Recipient** (*M:N*)
- **Hospital performs Transplant** (*M:N*)

29. Mental Health Counseling

A mental health platform connects patients and therapists for counseling.

Entities & Attributes:

- **Patient** (PatientID, Name, Age, ContactNumber, Email, MentalHealthHistory)
- **Therapist** (TherapistID, Name, Specialization, ContactNumber, Email)
- **Session** (SessionID, PatientID, TherapistID, Date, Duration, Notes)
- **Feedback** (FeedbackID, PatientID, TherapistID, Rating, Comments, Date)

Relationships:

- **Patient attends Session with Therapist** (*M:N*)
- **Patient provides Feedback for Therapist** (*M:N*)

30. Optical Store System

A store tracks customers, eye tests, and sales of glasses.

Entities & Attributes:

- **Customer** (CustomerID, Name, Age, ContactNumber, Email)
- **EyeTest** (TestID, CustomerID, OptometristID, Date, Result)
- **Optometrist** (OptometristID, Name, ContactNumber, Experience)
- **Glasses** (GlassesID, Brand, Type, Price, StockQuantity)
- **Purchase** (PurchaseID, CustomerID, GlassesID, Date, Price)

Relationships:

- **Customer undergoes EyeTest by Optometrist** (*M:N*)
- **Customer purchases Glasses** (*M:N*)

31. Bank Management System

Tracks customers, accounts, transactions, and loans.

Entities & Attributes:

- **Customer** (CustomerID, Name, ContactNumber, Email, Address)
- **Account** (AccountID, CustomerID, AccountType, Balance, OpenDate)
- **Transaction** (TransactionID, AccountID, Type, Amount, Date, Status)
- **Loan** (LoanID, CustomerID, Amount, InterestRate, StartDate, EndDate, Status)

Relationships:

- **Customer owns Account** (*1:M*)
- **Customer applies for Loan** (*M:N*)
- **Account has Transactions** (*1:M*)

32. Loan Management System

Handles loan applications, approvals, and repayments.

Entities & Attributes:

- **Applicant** (ApplicantID, Name, ContactNumber, Email, CreditScore)
- **Loan** (LoanID, ApplicantID, Type, Amount, InterestRate, Status)
- **Repayment** (RepaymentID, LoanID, AmountPaid, PaymentDate, RemainingBalance)

Relationships:

- **Applicant applies for Loan** (*M:N*)
- **Loan has Repayments** (*1:M*)

33. Credit Card System

Manages credit cards, spending limits, and payments.

Entities & Attributes:

- **Cardholder** (CardholderID, Name, ContactNumber, Email, Address)
- **CreditCard** (CardID, CardholderID, CardType, CreditLimit, ExpiryDate, Status)
- **Transaction** (TransactionID, CardID, Merchant, Amount, Date, Status)
- **Payment** (PaymentID, CardID, AmountPaid, PaymentDate)

Relationships:

- **Cardholder owns CreditCard** (1:M)
- **CreditCard has Transactions** (1:M)
- **Cardholder makes Payment** (M:N)

34. Stock Trading Platform

Tracks investors, stock prices, and trades.

Entities & Attributes:

- **Investor** (InvestorID, Name, ContactNumber, Email, PortfolioValue)
- **Stock** (StockID, Symbol, CompanyName, CurrentPrice, MarketCap)
- **Trade** (TradeID, InvestorID, StockID, Quantity, Price, TradeDate, Type)

Relationships:

- **Investor buys/sells Stock through Trade** (M:N)

35. Insurance Company

Manages policyholders, premium payments, and claims.

Entities & Attributes:

- **Policyholder** (PolicyholderID, Name, ContactNumber, Email, Age)
- **InsurancePolicy** (PolicyID, PolicyholderID, Type, CoverageAmount, PremiumAmount)
- **Claim** (ClaimID, PolicyID, AmountClaimed, Date, Status)
- **PremiumPayment** (PaymentID, PolicyID, Amount, PaymentDate)

Relationships:

- **Policyholder owns InsurancePolicy** (1:M)
- **Policyholder makes PremiumPayment** (M:N)
- **Policyholder submits Claim** (M:N)

36. Mutual Fund Investment

Tracks investment portfolios and returns.

Entities & Attributes:

- **Investor** (InvestorID, Name, ContactNumber, Email, RiskLevel)
- **MutualFund** (FundID, Name, NAV, FundManager, Type)
- **Investment** (InvestmentID, InvestorID, FundID, Amount, Date)

Relationships:

- **Investor invests in MutualFund** (M:N)

37. Mobile Wallet Service

Handles digital payments, transactions, and user balances.

Entities & Attributes:

- **User** (UserID, Name, ContactNumber, Email, Balance)
- **Transaction** (TransactionID, SenderID, ReceiverID, Amount, Date, Status)
- **Merchant** (MerchantID, Name, Category, ContactNumber)

Relationships:

- **User transfers money to another User** (M:N)
- **User makes payments to Merchant** (M:N)

38. ATM Transaction System

Manages cash withdrawals, deposits, and security.

Entities & Attributes:

- **ATM** (ATMID, Location, Bank, Status)
- **Cardholder** (CardholderID, Name, ContactNumber, CardNumber, Bank)
- **Transaction** (TransactionID, ATMID, CardholderID, Type, Amount, Date, Status)

Relationships:

- **Cardholder performs Transaction at ATM** (M:N)

39. Auction Investment Platform

Connects investors with startups through bidding.

Entities & Attributes:

- **Investor** (InvestorID, Name, ContactNumber, Email, AvailableFunds)
- **Startup** (StartupID, Name, Industry, FundingGoal)

- **Bid** (BidID, InvestorID, StartupID, Amount, Date, Status)

Relationships:

- **Investor places Bid on Startup** (*M:N*)

40. Cryptocurrency Exchange

Tracks users, wallets, and crypto transactions.

Entities & Attributes:

- **User** (UserID, Name, ContactNumber, Email, WalletBalance)
- **Wallet** (WalletID, UserID, CryptoType, Balance)
- **Transaction** (TransactionID, SenderWalletID, ReceiverWalletID, CryptoAmount, Date, Status)

Relationships:

- **User owns Wallet** (*1:M*)
- **Wallet processes Transaction** (*M:N*)

41. Social Networking Site

Tracks users, posts, likes, and comments.

Entities & Attributes:

- **User** (UserID, Name, Email, ContactNumber, ProfilePicture)
- **Post** (PostID, UserID, Content, Media, Timestamp)
- **Like** (LikeID, UserID, PostID, Timestamp)
- **Comment** (CommentID, UserID, PostID, Content, Timestamp)

Relationships:

- **User creates Post** (*1:M*)
- **User likes Post** (*M:N*)
- **User comments on Post** (*M:N*)

42. Music Streaming Service

Manages songs, artists, playlists, and subscriptions.

Entities & Attributes:

- **User** (UserID, Name, Email, SubscriptionType)
- **Artist** (ArtistID, Name, Genre)
- **Song** (SongID, ArtistID, Title, Duration, ReleaseDate)
- **Playlist** (PlaylistID, UserID, Name, CreatedDate)

Relationships:

- **Artist creates Song** (*1:M*)
- **User creates Playlist** (*1:M*)
- **Playlist contains Songs** (*M:N*)

43. Video Streaming Platform

Tracks movies, subscriptions, and reviews.

Entities & Attributes:

- **User** (UserID, Name, Email, SubscriptionType)
- **Movie** (MovieID, Title, Genre, ReleaseDate)
- **Review** (ReviewID, UserID, MovieID, Rating, Comments)

Relationships:

- **User watches Movie** (*M:N*)
- **User writes Review for Movie** (*M:N*)

44. Online Gaming Platform

Manages players, tournaments, and rewards.

Entities & Attributes:

- **Player** (PlayerID, Username, Rank, Score)
- **Game** (GameID, Title, Genre)
- **Tournament** (TournamentID, GameID, PrizePool, StartDate)

Relationships:

- **Player participates in Tournament** (*M:N*)
- **Player plays Game** (*M:N*)

45. Sports League System

Tracks teams, matches, and player statistics.

Entities & Attributes:

- **Team** (TeamID, Name, Coach, League)
- **Player** (PlayerID, Name, Age, Position, TeamID)
- **Match** (MatchID, Team1ID, Team2ID, Date, Score)

Relationships:

- **Team plays Match** ($M:N$)
- **Player belongs to Team** ($M:1$)

46. Film Production Company

Manages movies, actors, and production schedules.

Entities & Attributes:

- **Movie** (MovieID, Title, Genre, ReleaseDate)
- **Actor** (ActorID, Name, Role, MovieID)
- **ProductionSchedule** (ScheduleID, MovieID, StartDate, EndDate)

Relationships:

- **Actor acts in Movie** ($M:N$)

47. News Portal

Tracks articles, journalists, and reader comments.

Entities & Attributes:

- **Journalist** (JournalistID, Name, Email, Specialty)
- **Article** (ArticleID, JournalistID, Title, Content, PublishedDate)
- **Comment** (CommentID, UserID, ArticleID, Content, Timestamp)

Relationships:

- **Journalist writes Article** ($1:M$)
- **User comments on Article** ($M:N$)

48. Podcast Hosting

Manages podcasters, episodes, and subscriptions.

Entities & Attributes:

- **Podcaster** (PodcasterID, Name, ChannelName)
- **Episode** (EpisodeID, PodcasterID, Title, Duration)
- **User** (UserID, Name, Email, SubscriptionStatus)

Relationships:

- **Podcaster releases Episode** ($1:M$)
- **User subscribes to Podcaster** ($M:N$)

49. Event Management System

Tracks concerts, bookings, and attendees.

Entities & Attributes:

- **Event** (EventID, Name, Date, Venue)
- **Attendee** (AttendeeID, Name, Email)
- **Ticket** (TicketID, EventID, AttendeeID, Price)

Relationships:

- **Attendee buys Ticket for Event** ($M:N$)

50. Art Gallery Website

Manages artists, artworks, and exhibitions.

Entities & Attributes:

- **Artist** (ArtistID, Name, Bio)
- **Artwork** (ArtworkID, ArtistID, Title, Medium, Price)
- **Exhibition** (ExhibitionID, Name, Date, Location)

Relationships:

- **Artist creates Artwork** ($1:M$)
- **Artwork is displayed in Exhibition** ($M:N$)

51. Airline Reservation System

Tracks flights, passengers, and seat bookings.

Entities & Attributes:

- **Flight** (FlightID, Airline, Departure, Arrival, Date, Time)
- **Passenger** (PassengerID, Name, Email, ContactNumber)
- **Booking** (BookingID, FlightID, PassengerID, SeatNumber)

Relationships:

- **Passenger books Flight** (*M:N*)

52. Railway Reservation System

Manages train schedules and ticketing.

Entities & Attributes:

- **Train** (TrainID, Name, Route, DepartureTime)
- **Passenger** (PassengerID, Name, Email)
- **Ticket** (TicketID, TrainID, PassengerID, SeatNumber)

Relationships:

- **Passenger books Train Ticket** (*M:N*)

53. Tourism Company

Handles tour packages, travelers, and payments.

Entities & Attributes:

- **TourPackage** (PackageID, Destination, Duration, Cost)
- **Traveler** (TravelerID, Name, Email)
- **Booking** (BookingID, PackageID, TravelerID)

Relationships:

- **Traveler books TourPackage** (*M:N*)

54. Ride-Sharing App

Tracks drivers, riders, and payments.

Entities & Attributes:

- **Driver** (DriverID, Name, CarModel, Rating)
- **Rider** (RiderID, Name, ContactNumber)
- **Ride** (RideID, DriverID, RiderID, Fare, Distance)

Relationships:

- **Rider books Ride with Driver** (*M:N*)

55. Fleet Management

Manages vehicles, routes, and drivers.

Entities & Attributes:

- **Vehicle** (VehicleID, Model, RegistrationNumber)
- **Driver** (DriverID, Name, LicenseNumber)
- **Route** (RouteID, StartLocation, EndLocation)

Relationships:

- **Driver operates Vehicle** (*1:M*)

56. Bike Rental Service

Handles bike rentals and user data.

Entities & Attributes:

- **User** (UserID, Name, ContactNumber)
- **Bike** (BikeID, Model, Status)
- **Rental** (RentalID, UserID, BikeID, Duration, Cost)

Relationships:

- **User rents Bike** (*M:N*)

57. Logistics & Supply Chain

Tracks shipments and warehouses.

Entities & Attributes:

- **Shipment** (ShipmentID, Origin, Destination, Status)
- **Warehouse** (WarehouseID, Location, Capacity)

Relationships:

- **Warehouse stores Shipment** (*M:N*)

58. Visa Processing System

Manages applications and approvals.

Entities & Attributes:

- **Applicant** (ApplicantID, Name, PassportNumber)
- **VisaApplication** (ApplicationID, ApplicantID, Status, Type)

Relationships:

- **Applicant submits VisaApplication** (1:M)

59. Ferry Service

Tracks boats, schedules, and passengers.

Entities & Attributes:

- **Boat** (boat_id, name, capacity, status)
- **Schedule** (schedule_id, boat_id, route, departure_time, arrival_time, status)
- **Passenger** (passenger_id, name, contact, email)
- **Ticket** (ticket_id, passenger_id, schedule_id, seat_no, fare, booking_status)

Relationships:

- **Boat operates on Schedule** (1:M)
- **Passenger books Ticket** (M:N)
- **Ticket is assigned to Schedule** (M:1)

60. Metro Transit System

Manages train operations and passengers.

Entities & Attributes:

- **Train** (train_id, name, capacity, status)
- **Route** (route_id, start_station, end_station, distance)
- **Station** (station_id, name, location)
- **Schedule** (schedule_id, train_id, route_id, departure_time, arrival_time, status)
- **Passenger** (passenger_id, name, contact, email, ticket_type)
- **Ticket** (ticket_id, passenger_id, schedule_id, fare, ticket_status)

Relationships:

- **Train follows Route** (1:M)
- **Train operates on Schedule** (1:M)
- **Passenger purchases Ticket** (M:N)
- **Ticket belongs to Schedule** (M:1)

61. IoT-Based Smart Home

Manages smart devices and automation rules.

Entities & Attributes:

- **User** (user_id, name, email, password, home_id)
- **Home** (home_id, address, owner_id, total_devices)
- **Device** (device_id, home_id, type, status, last_updated)
- **AutomationRule** (rule_id, home_id, device_id, condition, action, status)
- **Log** (log_id, device_id, timestamp, action, value)

Relationships:

- **User owns Home** (1:M)
- **Home contains Devices** (1:M)
- **Device follows AutomationRule** (1:M)
- **Device generates Log** (1:M)

62. Blockchain-Based Voting

Tracks voters, elections, and votes.

Entities & Attributes:

- **Voter** (voter_id, name, national_id, public_key)
- **Election** (election_id, name, start_date, end_date, status)
- **Candidate** (candidate_id, election_id, name, party, manifesto, vote_count)
- **Vote** (vote_id, voter_id, election_id, candidate_id, timestamp, hash)
- **Blockchain** (block_id, previous_hash, current_hash, timestamp, transaction_data)

Relationships:

- **Voter participates in Election** (M:N)
- **Election has Candidates** (1:M)
- **Voter casts Vote** (1:1)
- **Vote is recorded in Blockchain** (1:M)

63. Job Portal

Connects job seekers with recruiters.

Entities & Attributes:

- **JobSeeker** (seeker_id, name, email, resume, skills, experience)
- **Employer** (employer_id, company_name, contact, industry, location)
- **Job** (job_id, employer_id, title, description, salary, location, status)
- **Application** (application_id, seeker_id, job_id, status, applied_date)
- **Interview** (interview_id, application_id, date, time, status)

Relationships:

- **Employer posts Job** (1:M)
- **JobSeeker applies for Job** (M:N)
- **Application leads to Interview** (1:M)

64. Smart Agriculture System

Tracks farm sensors and weather data.

Entities & Attributes:

- **Farm** (farm_id, name, location, owner_id)
- **Farmer** (farmer_id, name, contact, farm_id)
- **Sensor** (sensor_id, farm_id, type, status, last_reading)
- **WeatherData** (weather_id, farm_id, temperature, humidity, rainfall, timestamp)
- **IrrigationLog** (log_id, farm_id, timestamp, duration, status)

Relationships:

- **Farmer owns Farm** (1:M)
- **Farm has Sensors** (1:M)
- **Sensor records WeatherData** (1:M)
- **Farm maintains IrrigationLog** (1:M)

Student	Q1	Q2	Q3
713523AM060	3	15	32
713523AM061	8	21	45
713523AM062	12	27	51
713523AM063	5	18	38
713523AM064	2	29	62
713523AM065	7	24	49
713523AM066	10	19	40
713523AM067	6	13	57
713523AM068	14	25	34
713523AM069	1	20	60
713523AM071	9	22	46
713523AM072	11	26	55
713523AM073	4	30	36
713523AM074	16	17	42
713523AM075	28	23	48
713523AM076	5	10	53
713523AM077	18	8	39
713523AM078	21	12	50
713523AM079	27	2	63
713523AM080	29	7	33
713523AM081	25	14	43

713523AM082	6	17	61
713523AM083	20	9	47
713523AM084	11	22	52
713523AM085	1	26	56
713523AM086	30	16	41
713523AM087	3	28	35
713523AM088	24	13	44
713523AM089	19	15	58
713523AM090	2	5	59
713523AM091	10	21	37
713523AM093	12	27	64
713523AM094	8	29	31
713523AM095	7	18	54
713523AM096	14	25	32
713523AM097	9	1	50
713523AM098	5	17	39
713523AM099	20	28	42
713523AM100	3	26	55
713523AM101	23	16	48
713523AM102	2	30	46
713523AM103	21	9	36
713523AM104	10	12	33
713523AM105	4	19	63
713523AM106	22	6	41
713523AM107	15	24	37
713523AM108	7	1	52
713523AM109	14	20	58
713523AM110	27	8	34
713523AM111	11	5	62
713523AM112	3	29	51
713523AM113	16	13	44
713523AM114	26	17	40
713523AM115	2	22	60
713523AM116	30	9	45
713523AM117	10	12	53
713523AM507	23	4	35
713523AM508	6	28	64
713523AM509	25	1	38
713523AM510	14	19	57
713523AM511	11	15	49
713523AM512	5	21	47

Deadline: 14.03.2025

Faculty Incharge

HoD