

ER-Diagrams HW2

Query Masters

2025-02-10

GITHUB URL: [GitHub Repository](#)

Group Members

Names AND V#

- Enoch Kumanyenu V01114695
- Rana Nolak V01007582

Introduction

This report presents the design of three database systems using Chen and Crow's Foot ER diagrams. The systems chosen for this assignment are the **Sports Tournament Management System**, **Real Estate Listing System**, and **Hotel Booking System**.

Sports Tournament Management System

Entity-Relationship Diagrams

Chen Diagram

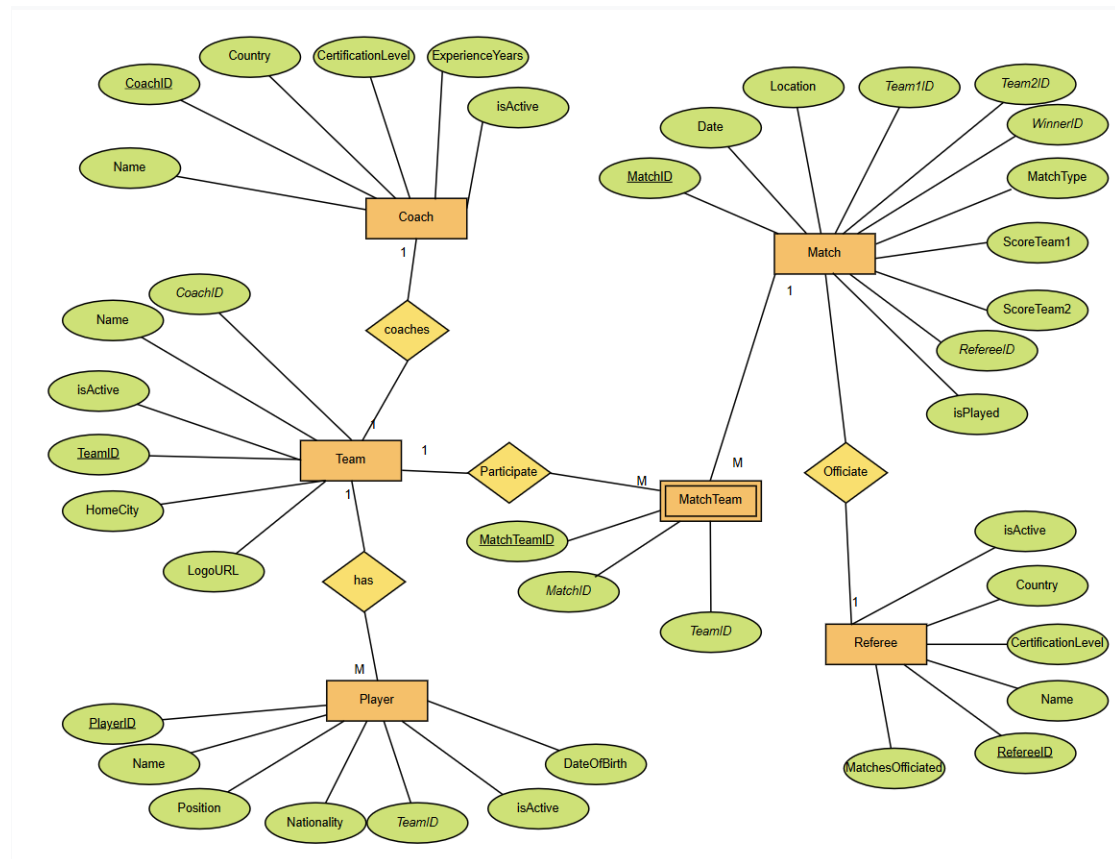


Figure 1: Chen Diagram

Crow's Foot Diagram

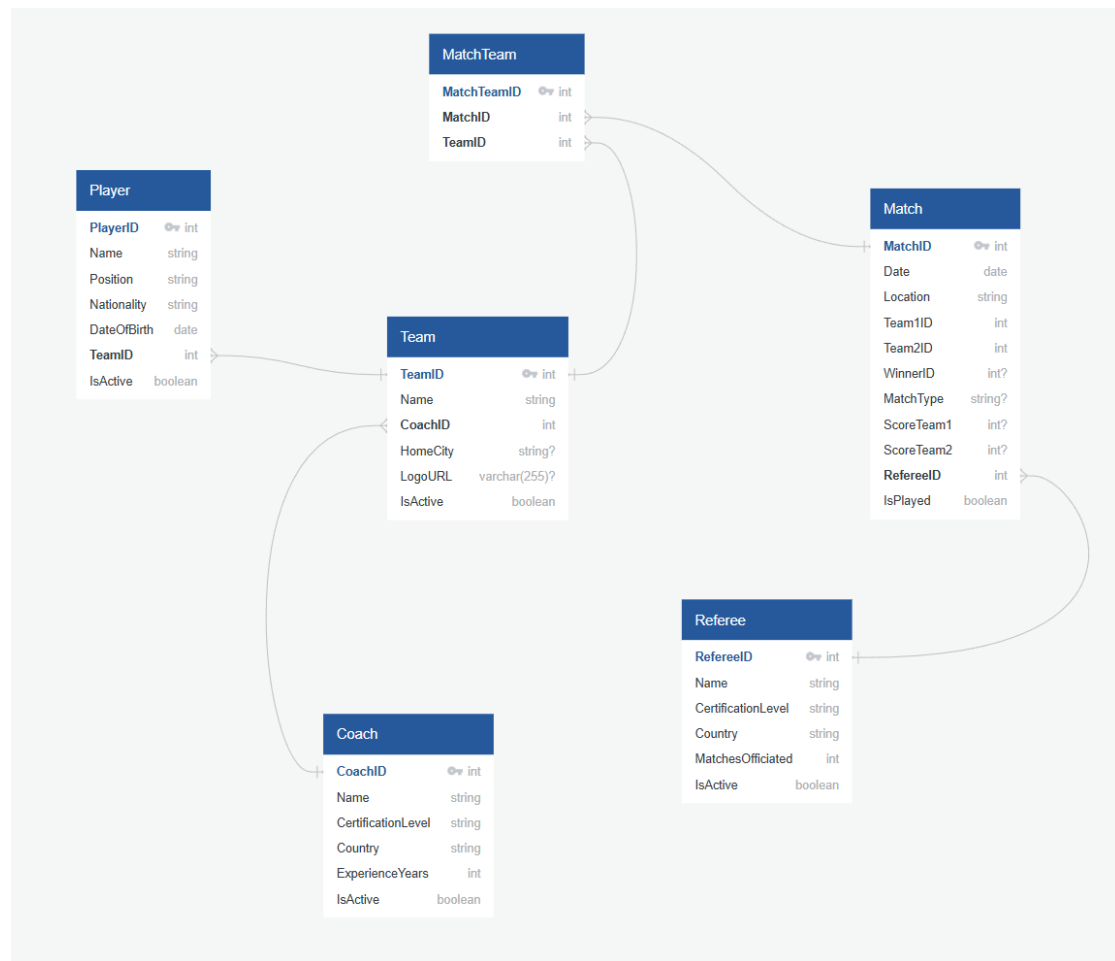


Figure 2: Crow's Foot Diagram

Relation Set Schema

- **Team** (TeamID(PK), Name, CoachID(FK), HomeCity, LogURL)
- **Player** (PlayerID(PK), Name, TeamID(FK), Position, Nationality, DateOfBirth)
- **Match** (MatchID(PK), Date, Location, Team1ID(FK), Team2ID(FK), WinnerID, RefereeID(FK))
- **Coach** (CoachID(PK), Name, CertificationLevel, Country, ExperienceYears)

- **Referee** (RefereeID(PK), Name, Country, CertificationLevel, ExperienceYears,MatchesOfficiated)

Design Decisions

- **Weak Entities:** The MatchTeam entity is a weak entity as it relies on both Match and Team for its existence.
- **Composite Attributes:** Player names are kept simple; first and last names are stored as a single attribute.
- **Multivalued Attributes:** Each team has a single coach, so no multivalued attributes were introduced.

Real Estate Listing System

Entity-Relationship Diagrams

Chen Diagram

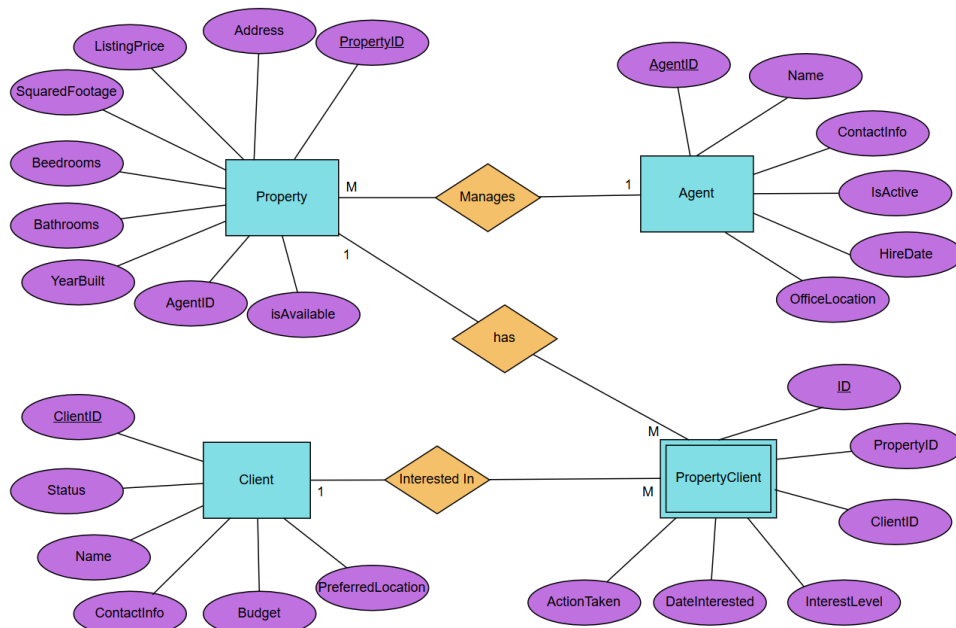


Figure 3: Chen Diagram

Crow's Foot Diagram

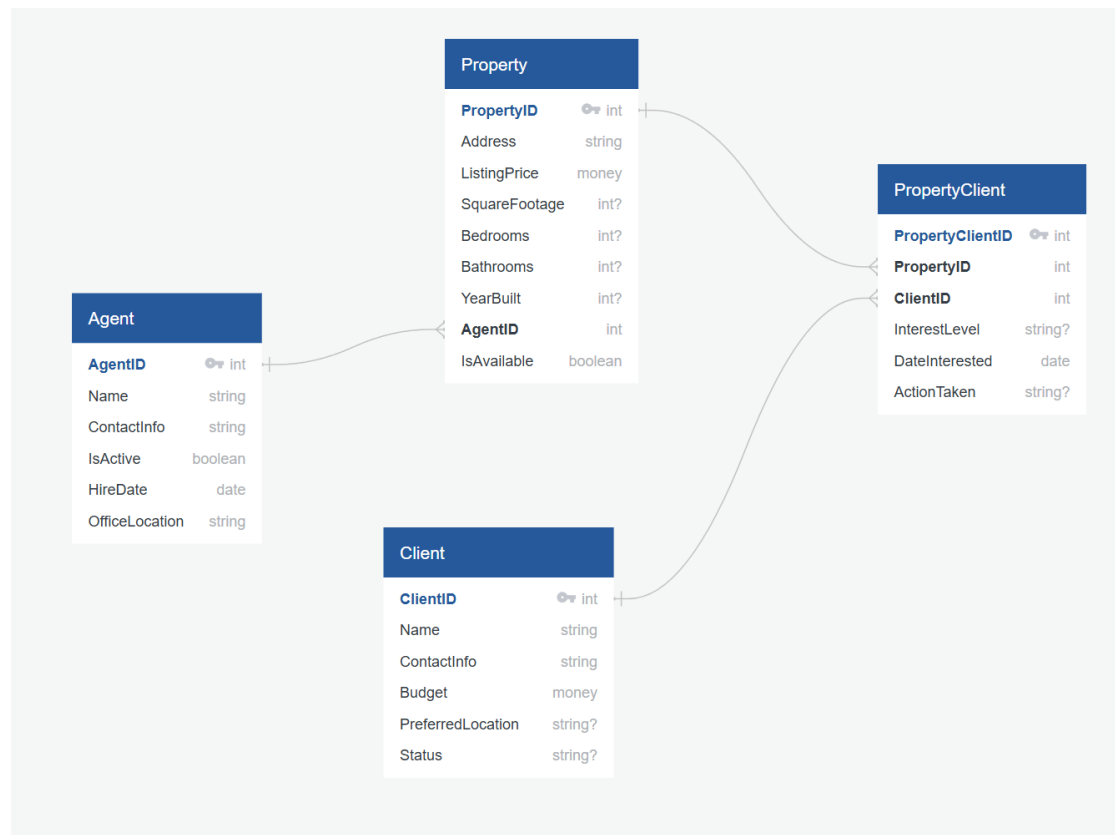


Figure 4: Crow's Foot Diagram

Relation Set Schema

- **Property** (PropertyID(PK), Address, ListingPrice, Bedrooms, Bathrooms, YearBuilt, AgentID(FK), IsAvailable)
- **Agent** (AgentID(PK), Name, ContactInfo, IsActive, HireDate, OfficeLocation)
- **Client** (ClientID(PK), Name, ContactInfo, Budget, PreferredLocation, Status)
- **PropertyClient** (PropertyClientID(PK), PropertyID(FK), ClientID(FK), InterestLevel, DateInterested, ActionTaken)

Design Decisions

- **Weak Entities:** The PropertyClient entity is a weak entity as it relies on both Property and Client for its existence.
- **Composite Attributes:** Names are not split into first and last names; instead, they are stored as single strings.
- **Multivalued Attributes:** Clients can be interested in multiple properties, which is represented through the PropertyClient relationship.

Hotel Booking System

Entity-Relationship Diagrams

Chen Diagram

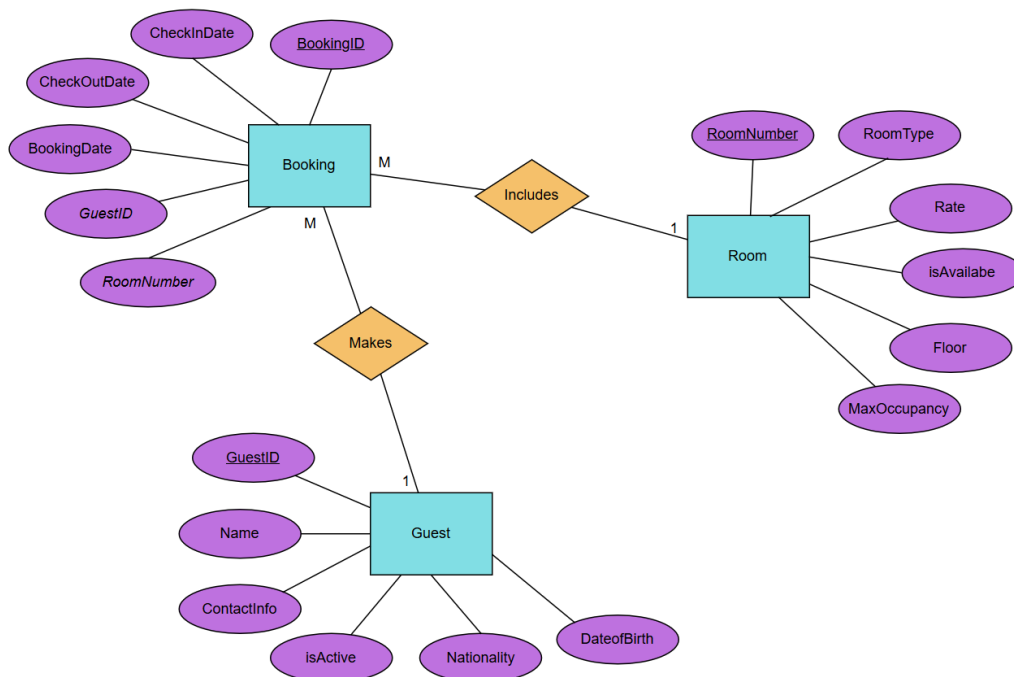


Figure 5: Chen Diagram

Crow's Foot Diagram

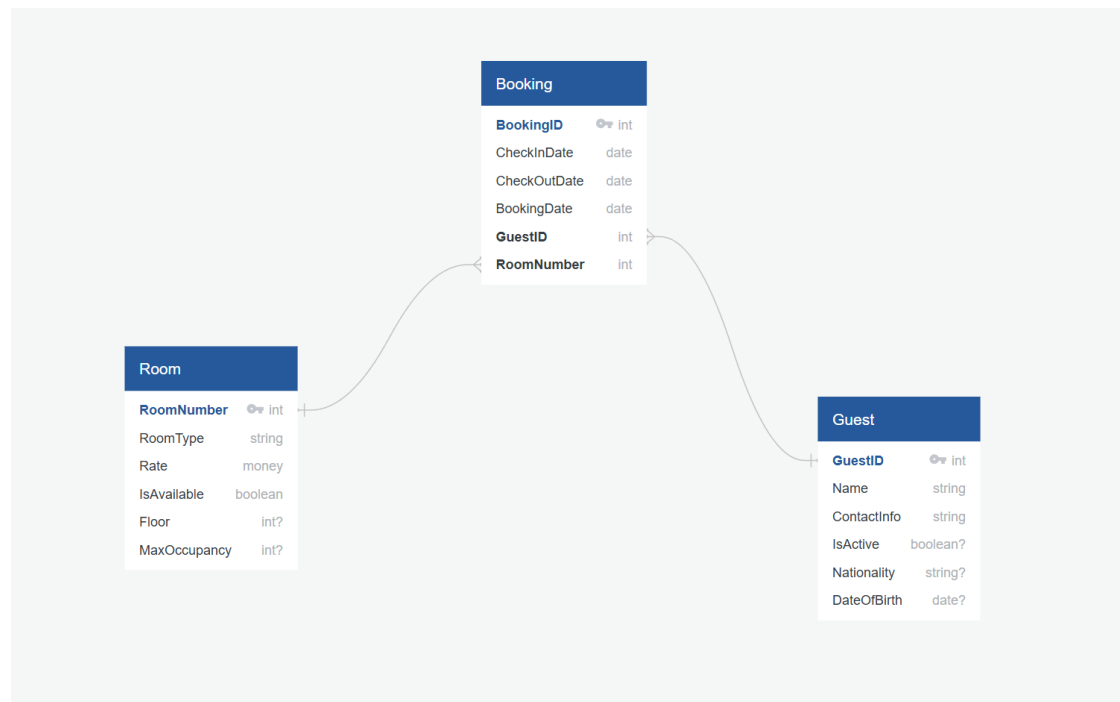


Figure 6: Crow's Foot Diagram

Relation Set Schema

- **Room** (RoomNumber(PK), RoomType, Rate, IsAvailable, Floor, MaxOccupancy)
- **Guest** (GuestID(PK), Name, ContactInfo, IsActive, Nationality, DateOfBirth)
- **Booking** (BookingID(PK), CheckInDate, CheckOutDate, GuestID(FK), RoomNumber(FK))

Design Decisions

- **Weak Entities:** No weak entities were introduced, as all entities can operate independently.
- **Composite Attributes:** Guest names are stored as single attributes without composite breakdown.

- **Multivalued Attributes:** The system assumes that each room is uniquely assigned to a booking, preventing multiple bookings for the same room at the same time.

Design Overview

This section provides an overview of how the entities and relationships were structured in each system. By analyzing the requirements of each system and identifying key entities, we were able to create efficient and normalized database designs.

- **Normalization:** All schemas were designed to minimize redundancy and ensure data integrity.
- **Entity Relationships:** Relationships were carefully defined to reflect real-world interactions, ensuring that all necessary relationships were captured.

Conclusion

This report outlined the design process for three distinct database systems. Each system's design was informed by the specific requirements and relationships inherent to its domain. The use of ER diagrams in both Chen and Crow's Foot notations allowed for a clear visual representation of entities and relationships, while the relational schemas ensured proper normalization and integrity across the databases. The design choices made reflect a balance between simplicity and functionality, aiming to create efficient and effective database systems.