DATABASE SYSTEMS

Project Title:

Real Estate Property Sale Management System

Introduction

This project focuses on the design and implementation of a relational database system for a real estate company specializing in the sale of residential properties. The company operates through multiple agencies, employs agents, and deals with clients who are interested in purchasing properties owned by various owners.

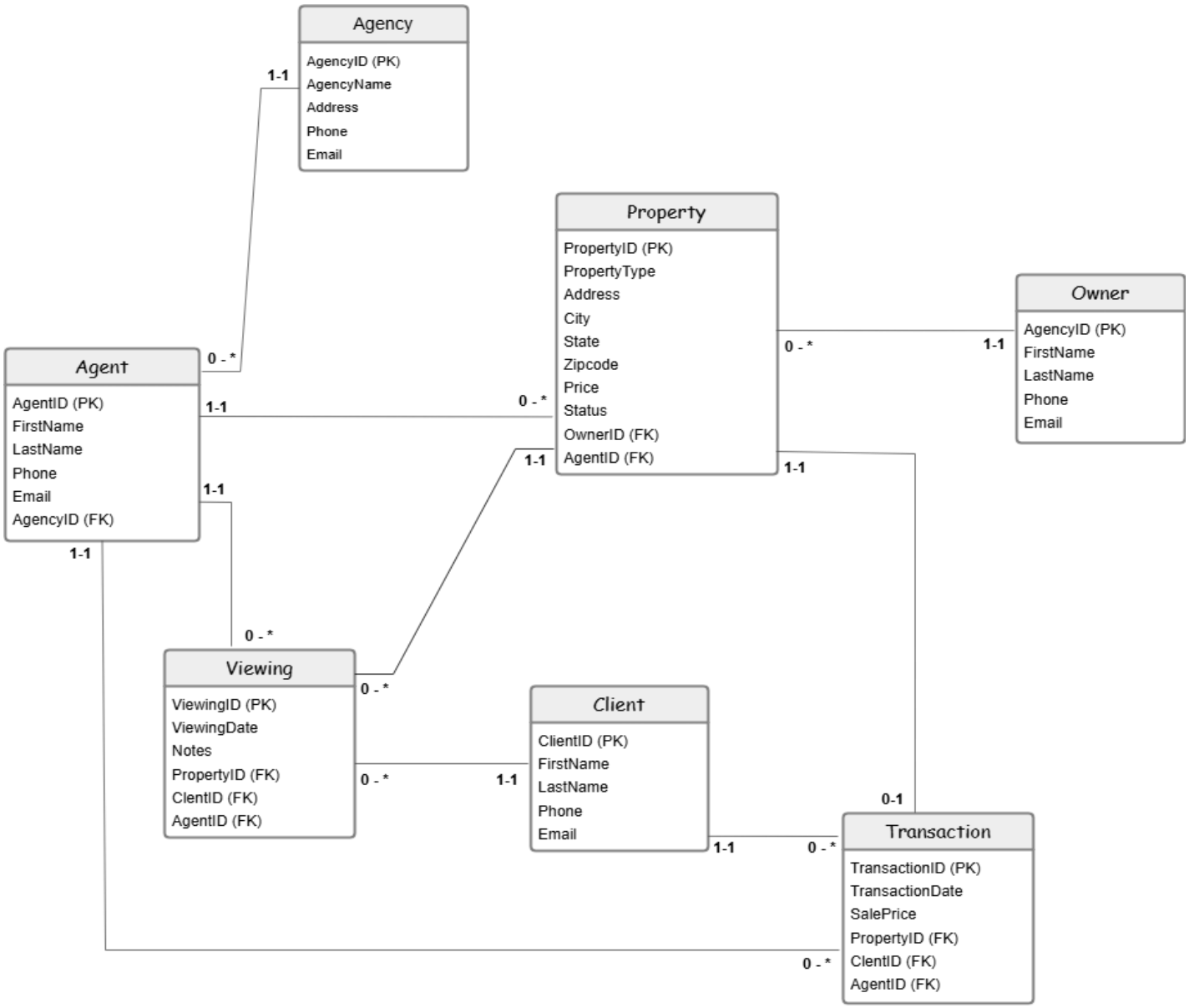
The database will help the company to efficiently manage and track:

- Details of properties, their owners, and buyers
- Assignments of agents to properties
- Management of agencies and their employees
- Property viewings scheduled between agents and clients
- Records of sales transactions

The goal of this database is to ensure:

- Referential integrity across all entities and relationships
- All relationships are simple, with no many-to-many relationships
- Clear and easy-to-use structure for both company employees and database administrators

UML ERD Diagram



Assumptions

1. Every **property** is assigned to exactly **one agent**, who is responsible for its sale process.
2. A **client** can purchase multiple properties, but each property is sold only once.
3. The company exclusively handles **property sales**, not rentals or leases.
4. Assume agents receive salaries or commissions through their agency (the database does not track commissions directly) and thus agency maintain its reputation.
5. **Properties** are sold from **owners to clients** (buyers) — agents facilitate the transaction but do not own the properties.
6. The **status** attribute of a property shows whether it is **available** or **sold.**
7. The **state** attribute of a property shows which province.
8. Agencies do not own properties directly; ownership always belongs to individuals (owners/clients).

Relational Schema

1. Owner

Owner (OwnerID, FirstName, LastName, Phone, Email)

Primary Key OwnerID

2. Agency

Agency (AgencyID, AgencyName, Address, Phone, Email)

Primary Key AgencyID

3. Agent

Agent (AgentID, FirstName, LastName, Phone, Email, AgencyID)

Primary Key AgentID

Foreign Key AgencyID **references** Agency(AgencyID) **On Delete No Action On Update Cascade**

4. Client

Client (ClientID, FirstName, LastName, Phone, Email)
Primary Key ClientID

5. Property

Property (PropertyID, PropertyType, Address, City, State, ZipCode, Price, Status, OwnerID, AgentID)
Primary Key PropertyID
Foreign Key OwnerID **references** Owner(OwnerID) **On Delete No Action On Update Cascade**
Foreign Key AgentID **references** Agent(AgentID) **On Delete No Action On Update Cascade**

6. Viewing

Viewing (ViewingID, ViewingDate, Notes, PropertyID, ClientID, AgentID)
Primary Key ViewingID
Foreign Key PropertyID references Property(PropertyID) **On Delete No Action On Update Cascade**
Foreign Key ClientID references Client(ClientID) **On Delete No Action On Update Cascade**
Foreign Key AgentID references Agent(AgentID) **On Delete No Action On Update Cascade**

7. Transaction

Transaction (TransactionID, TransactionDate, SalePrice, PropertyID, ClientID, AgentID)
Primary Key TransactionID
Foreign Key PropertyID references Property(PropertyID) **On Delete No Action On Update Cascade**
Foreign Key ClientID references Client(ClientID) **On Delete No Action On Update Cascade**
Foreign Key AgentID references Agent(AgentID) **On Delete No Action On Update Cascade**

Relationship	Mandatory?	Reason
Owner → Property	YES	Needed to know who owns the property
Agent → Property	YES	Needed to know who handles the property
Agency → Agent	YES	Needed to know which agency the agent works for
Property → Viewing	YES	Needed to track which property is viewed
Client → Viewing	YES	Needed to track which client is viewing
Agent → Viewing	YES	Needed to track which agent conducted the viewing
Property → Transaction	YES	Needed to know which property was sold
Client → Transaction	YES	Needed to know who bought the property
Agent → Transaction	YES	Needed to know who handled the transaction