

Database Design for a Real Estate Office DBMS

1. Introduction

This document outlines the initial database design for a Real Estate Office Database Management System (DBMS). The system will support the sale and rental of properties within a city, managing information for Agents, Owners, and Administrators. It will store detailed property data and transaction records for sales and rentals, facilitating queries and performance analysis. This design is intended to support initial testing with 10–20 properties.

2. Key Assumptions

- **Property Listings:** Properties can be listed for sale, rent, or both, indicated by availability flags.
- **Transaction Data:** Sales and rental transactions are recorded in separate tables, including transaction details, dates, and market duration. Historical owner data is preserved in transaction tables, even if current ownership changes.
- **User Roles:**
 - **Agent:** Manages property transactions.
 - **Owner:** Owns properties.
 - **Admin:** Manages the database system.
- **Data Volume:** Designed for testing with 10–20 properties.
- **Normalization:** The design adheres to 1NF, 2NF, and 3NF to minimize redundancy and ensure data integrity.

3. Entity-Relationship Overview

- **Agent:** Manages multiple Sales and Rental transactions.
- **Owner:** Owns one or more Properties.
- **Admin:** Manages the database system.
- **Property:** Contains detailed property information and is associated with one Owner.
- **Sales:** Records details of each sale transaction, linking Agent, Property, and Owner.
- **Rental:** Records details of each rental transaction, linking Agent, Property, and Owner.

4. Relational Schema

4.1. Agent

Attribute	Data Type	Constraints
AgentID	INT ▾	PRIMARY KEY, AUTO_... ▾
Name	VARCHAR(100) ▾	NOT NULL ▾
Contact	VARCHAR(15) ▾	NOT NULL ▾

Email	VARCHAR(100) ▾	NOT NULL ▾
-------	----------------	------------

4.2. Owner

Attribute	Data Type	Constraints
OwnerID	INT ▾	PRIMARY KEY, AUTO_... ▾
Name	VARCHAR(100) ▾	NOT NULL ▾
Contact	VARCHAR(15) ▾	NOT NULL ▾
Email	VARCHAR(100) ▾	NOT NULL ▾

4.3. Admin

Attribute	Data Type	Constraints
AdminID	INT ▾	PRIMARY KEY, AUTO_INCREMENT
Username	VARCHAR(100) ▾	NOT NULL, UNIQUE
Password	VARCHAR(100) ▾	NOT NULL

4.4. Property

Attribute	Data Type	Constraints
PropertyID	INT ▾	PRIMARY KEY, AUTO_... ▾
OwnerID	INT ▾	FOREIGN KEY REFER... ▾
Address	VARCHAR(255) ▾	NOT NULL ▾
City	VARCHAR(100) ▾	NOT NULL ▾
Locality	VARCHAR(100) ▾	NOT NULL ▾
Size_sqft	INT ▾	NOT NULL ▾
Bedrooms	INT ▾	NOT NULL ▾

YearOfConstruction	YEAR ▾	NOT NULL ▾
SellingPrice	DECIMAL(12, 2) ▾	NULL (if not for sale) ▾
RentAmount	DECIMAL(12, 2) ▾	NULL (if not for rent) ▾
IsAvailableForSale	BOOLEAN ▾	NOT NULL ▾
IsAvailableForRent	BOOLEAN ▾	NOT NULL ▾

4.5. Sales

Attribute	Data Type	Constraints
SalesID	INT ▾	PRIMARY KEY, AUTO_... ▾
PropertyID	INT ▾	FOREIGN KEY REFER... ▾
AgentID	INT ▾	FOREIGN KEY REFER... ▾
OwnerID	INT ▾	FOREIGN KEY REFER... ▾
SalePrice	DECIMAL(12, 2) ▾	NOT NULL ▾
SaleDate	DATE ▾	NOT NULL ▾
MarketTime	INT ▾	(in days) NOT NULL ▾

4.6. Rental

Attribute	Data Type	Constraints
RentalID	INT ▾	PRIMARY KEY, AUTO_... ▾
PropertyID	INT ▾	FOREIGN KEY REFER... ▾
AgentID	INT ▾	FOREIGN KEY REFER... ▾
OwnerID	INT ▾	FOREIGN KEY REFER... ▾

RentAmount	DECIMAL(12, 2) ▾	NOT NULL ▾
RentalStartDate	DATE ▾	NOT NULL ▾
RentalEndDate	DATE ▾	NULL (if ongoing) ▾
MarketTime	INT ▾	(in days) NOT NULL ▾

5. Normalization

- **1NF:** Each table contains only atomic values and a primary key.
- **2NF:** All non-key attributes fully depend on the primary key.
- **3NF:** No transitive dependencies; each table represents a single subject area.

6. Conclusion

This initial database design provides a foundation for a Real Estate Office DBMS. It supports various user roles and detailed property and transaction information. Future steps include populating tables with test data and developing interfaces to support queries and reports. Adjustments may be made during implementation and testing.