**REAL ESTATE DATABASE MODEL DOCUMENTATION**

**Business Description**

The Real Estate model is designed for managing real estate properties, client details, agent assignments, transactions, and related financial records. The business involves the sale and rental of properties, where the real estate auction house facilitates these transactions by connecting sellers, buyers, landlords, tenants, and real estate agents.

**The database will track:**

Properties: Details about real estate properties such as their name, description, size, price, location, and type (sale or rental).

Clients: Information on clients who are either buyers, sellers, landlords, or tenants, including personal contact details.

Agents: Agents assigned to manage properties and facilitate transactions.

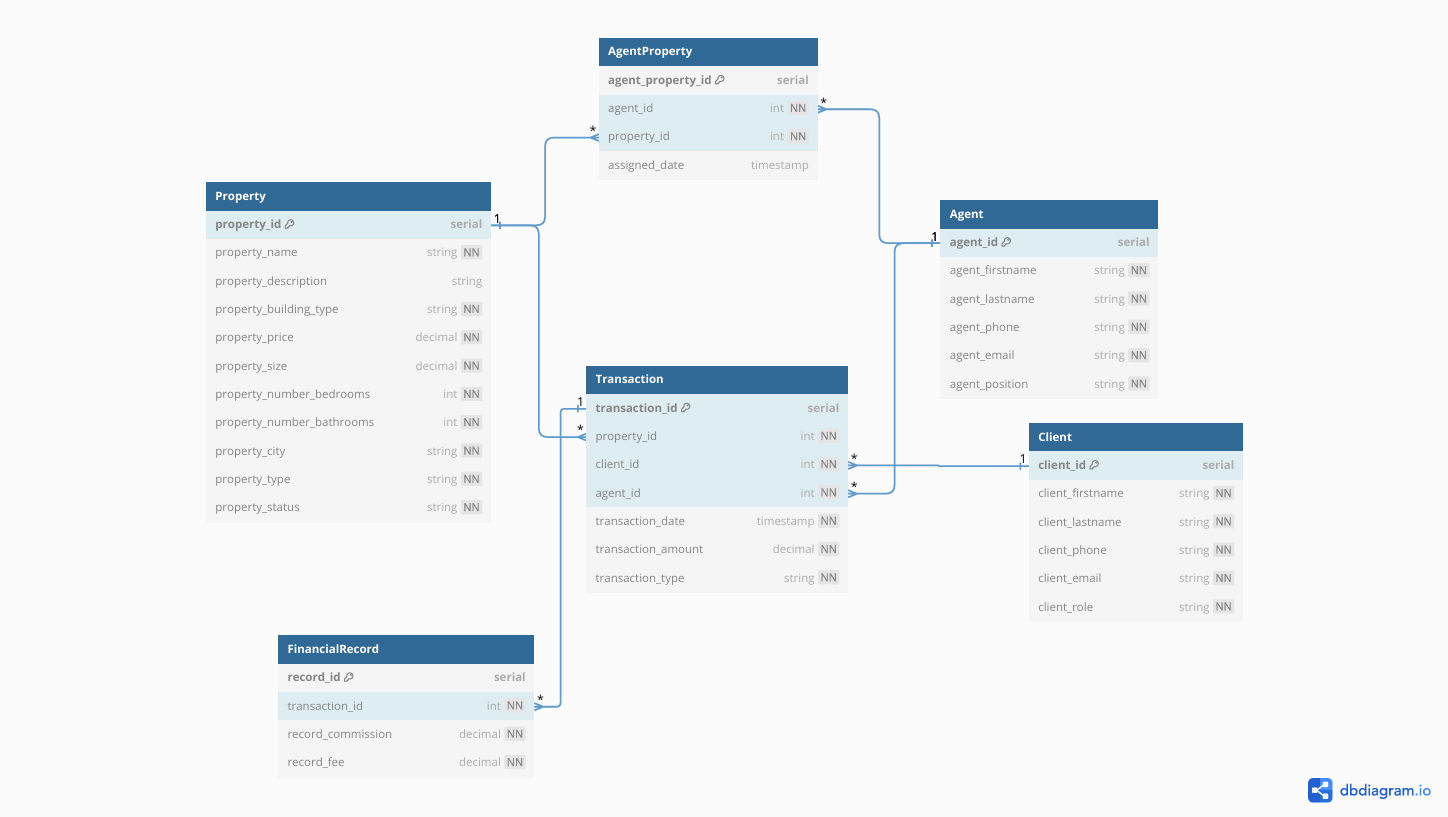
Transactions: Records of sale or rental transactions involving properties, clients, and agents.

Financial Records: Commissions and fees associated with each transaction, detailing the earnings of agents and auction house.

Agent Assignments: Tracking which agent is assigned to manage a particular property.

This model is designed to help the auction house manage the relationship between the agents, clients, and properties, while also ensuring proper tracking of transactions and commissions.

**Step-by-Step Approach to Modeling**



More constrains you can see here <https://dbdiagram.io/d/6749f952e9daa85aca2a459f>

**1.** Identifying the Entities

The primary entities involved in the business were identified as:

Property: Represents the real estate properties being sold or rented.

Client: Represents individuals or organizations who are buying, selling, renting, or leasing properties.

Agent: Represents real estate agents facilitating the transactions.

Transaction: Represents the sale or rental of properties.

FinancialRecord: Represents the financial transactions that include agent commissions and other fees.

AgentProperty: Represents the assignment of agents to properties.

**2.** Defining Tables and Columns

Each of the identified entities was translated into a table with appropriate columns.

* Property Table:

Captures details like the property name, description, type (house or apartment), price, size, number of bedrooms, bathrooms and the status (available or sold).

Generated Columns: Property size is converted from square feet to square meters automatically using a generated column.

* Client Table:

Contains client information such as name, phone number, email, and their role (buyer, seller, landlord, or tenant).

Email and Phone Constraints: Ensured valid email and phone number formats through unique constraints.

* Agent Table:

Contains information about the agents, including their names, phone numbers, emails, and position (sales agent, support staff, etc.).

Full Name: A generated column concatenates the agent's first and last name for easier querying.

* Transaction Table:

Tracks property transactions, including the transaction amount and type (sale or rental).

Transaction Date: Set to the current timestamp by default.

Generated Column: Calculates the transaction amount after deducting agent commissions.

* FinancialRecord Table:

Captures commission and fees associated with each transaction.

Includes a relationship to the Transaction table via a foreign key.

* AgentProperty Table:

Associates agents with properties they are responsible for.

Each entry tracks which agent is assigned to a property and the date of assignment.

**3.** Defining Relationships

The relationships between the entities were established using foreign keys:

Transaction Table: Links to Property, Client, and Agent via foreign keys.

FinancialRecord Table: Links to Transaction.

AgentProperty Table: Links to both Agent and Property tables.

These relationships ensure referential integrity and provide a way to link transactions, agents, properties, and clients together.

**4.** Applying Constraints

Various constraints were applied to ensure data integrity:

Primary Keys: Ensured each entity has a unique identifier (e.g., property\_id, client\_id, etc.).

Foreign Keys: Enforced relationships between tables, ensuring that data in dependent tables (like Transaction, FinancialRecord, and AgentProperty) must correspond to entries in the referenced tables.

Check Constraints:

Ensured transaction amounts and property sizes are non-negative.

Validated phone numbers and email addresses using regex patterns.

Ensured that transaction dates are not before a specific date (e.g., July 1, 2024).

**5.** Default Values and Generated Columns

Generated Columns: Automatically calculated values like property\_area\_m2 for properties and transaction\_amount\_after\_fee for transactions.

Default Values: Used defaults for columns like transaction\_date and assigned\_date, which automatically get the current timestamp if not explicitly provided.

**6.** Populating Sample Data

The tables were populated with sample data to represent real-life transactions:

Properties: A range of properties, both for sale and for rent.

Clients: Different clients, including buyers, sellers, landlords, and tenants.

Agents: A mix of agents handling different types of properties.

Transactions: Sample sale and rental transactions.

Financial Records: Commissions and fees associated with the transactions.

Agent Assignments: Agents assigned to specific properties.

**Conclusion**

This database model ensures a comprehensive and robust management system for a real estate auction house, facilitating the tracking of properties, clients, agents, transactions, and commissions. With the application of constraints, default values, and generated columns, the system ensures data integrity, prevents erroneous data, and automates some calculations.

The design is flexible, scalable, and supports essential business processes such as:

Managing property listings (both sale and rental).

Facilitating client-agent-property interactions.

Handling financial records related to transactions.

By enforcing strong data integrity through constraints and automating calculations, the model aims to minimize errors and streamline operations for the auction house.