

|  |
| --- |
| Business Template  **Real estate agency** |
|  |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## Business background

Zillow, founded in 2006, has established itself as one of the leading real estate marketplaces in the United States. Its mission is to simplify and enhance the process of buying, selling, renting, and financing properties through innovative technology and user-friendly platforms. Zillow serves millions of users daily, offering them access to an extensive database of property listings, market trends, and analytical tools. The company is renowned for its Zestimate, an algorithm-based tool that predicts property values with remarkable accuracy, providing users with valuable insights into the housing market.

Zillow's platform connects a wide range of stakeholders, including buyers, sellers, agents, and renters. The company aggregates property data from various sources to offer a comprehensive view of the real estate market. By leveraging technology, Zillow enables users to explore properties, calculate mortgages, and even pre-qualify for financing. With over 135 million properties listed on its platform, Zillow is not only a trusted marketplace but also a data powerhouse in the real estate sector.

## Problems. Current Situation

Despite its success, Zillow faces challenges that stem from the scale and complexity of its operations. One of the most pressing issues is data overload. The platform handles vast quantities of property-related information, including images, prices, transaction histories, and geographic data. Managing this data efficiently while ensuring accuracy and relevance is a significant hurdle. Another challenge is the inefficiency in retrieving data quickly. Given the immense volume of users searching for properties simultaneously, delays in delivering search results can impact the user experience.

Additionally, Zillow struggles with data inconsistency. Duplicate or outdated property listings sometimes appear on the platform, which can lead to customer dissatisfaction and a diminished sense of trust in the platform’s reliability. As Zillow's user base continues to grow, the need for scalability becomes more critical. Supporting millions of concurrent users and managing real-time property updates require a robust and flexible system.

## the Benefits of implementing a database. Project Vision

Implementing a well-structured database tailored to Zillow's operational needs could provide transformative solutions to these challenges. A centralized database system would offer a structured way to organize property, client, and agent data, reducing redundancy and improving data consistency. With optimized query structures and indexing, Zillow could significantly enhance its search performance, ensuring that users receive timely and accurate search results.

Scalability would also be greatly improved with a modern database capable of handling the platform’s increasing demands. By leveraging horizontal scaling techniques, Zillow can ensure that its services remain fast and reliable, even during peak traffic periods. Real-time updates to property statuses could be achieved through advanced features like triggers and data syncing, ensuring users always see the most accurate and relevant information. This improvement would enhance the user experience by empowering buyers, renters, and agents to make informed decisions more efficiently.

A robust database system would also support advanced analytics, enabling Zillow to refine its tools, such as the Zestimate, and offer personalized recommendations to users. This capability would allow the company to remain competitive by delivering cutting-edge insights and value to its audience. Furthermore, integrating APIs and third-party tools with the database would streamline workflows, improving communication between agents and clients and fostering smoother transactions.

In conclusion, implementing a modern database for Zillow would not only address its current challenges but also pave the way for sustained growth and innovation. By enhancing data organization, performance, scalability, and user experience, the database would strengthen Zillow's position as a leader in the real estate market. This strategic investment would enable the company to provide unparalleled services to its users while maintaining the trust and satisfaction of millions who rely on its platform.

# Model description

## Definitions & Acronyms

Real Estate Agency: An organization that manages property listings, facilitates property transactions, and connects buyers or renters with sellers or landlords.

Agent: A professional representing the agency, managing properties, and assisting clients.

Client: A buyer, renter, seller, or landlord interacting with the agency.

Property: A real estate listing available for sale or rent.

Transaction: A completed sale or rental agreement between a client and the agency.

Visit: A scheduled appointment by a client to view a property.

Payment: A monetary record associated with a transaction.

## Logical Scheme

A diagram of a computer

Description automatically generated

## Objects

Agent

Represents a real estate agent who manages property listings and helps clients buy or rent properties.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type | Constraints |
| Agent | agent\_id | Unique ID for each agent | INTEGER | Primary Key (PK) |
| agent\_name | Full name of the agent | VARCHAR(100) | NOT NULL |
|  | email | Agent's email address | VARCHAR(150) | UNIQUE, NOT NULL |
|  | phone | Contact phone number | VARCHAR(20) | NOT NULL |

Comments on table relationships

Agents and Properties have a one-to-many relationship. Each property is managed by one agent, but an agent can manage multiple properties.

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **agent\_id** | | |  | | --- | | **agent\_id** | | |  | | --- | | **agent\_id** | | |  | | --- | | **agent\_id** | |
| 1 | John Doe | john.doe@agency.com | 555-1111111 |
| 2 | Jane Smith | jane.smith@agency.com | 555-2222222 |
| 3 | Sarah Taylor | sarah.taylor@agency.com | 555-3333333 |

Client

Represents a buyer, renter, or seller interacting with the real estate agency.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type | Constraints |
| Client | client\_id | Unique ID for each client | INTEGER | Primary Key (PK) |
| client\_name | Full name of the client | VARCHAR(100) | NOT NULL |
|  | email | Client's email address | VARCHAR(150) | UNIQUE, NOT NULL |
|  | phone | Contact phone number | VARCHAR(20) | NOT NULL |

Comments on table relationships

Clients and Properties have a many-to-many relationship, resolved through the Transactions table. A client can transact on multiple properties, and a property can have multiple transactions over time.

Clients and Properties also have a many-to-many relationship through the Visits table. A client may visit multiple properties, and a property may have multiple visits from different clients.

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **client\_id** | | |  | | --- | | **client\_name** | | |  | | --- | | **email** | | |  | | --- | | **phone** | |
| 1 | Alice Johnson | alice.johnson@mail.com | 555-4444444 |
| 2 | Bob Brown | bob.brown@mail.com | 555-5555555 |
| 3 | Carol Davis | carol.davis@mail.com | 555-6666666 |

Properties

Represents a real estate listing available for sale or rent.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type | Constraints |
| Properties | property\_id | Unique ID for each property | INTEGER | Primary Key (PK) |
| address | Property address | VARCHAR(200) | NOT NULL |
|  | price | Listing price of the property | DECIMAL(10,2) | CHECK price > 0 |
|  | agent\_id | Linked agent managing the property | INT | Foreign Key (FK), NOT NULL |

Comments on table relationships

Agents and Properties have a one-to-many relationship. Each property is managed by one agent, but an agent can manage multiple properties.

Clients and Properties have a many-to-many relationship, resolved through the Transactions table. A client can transact on multiple properties, and a property can have multiple transactions over time.

Clients and Properties also have a many-to-many relationship through the Visits table. A client may visit multiple properties, and a property may have multiple visits from different clients.

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **property\_id** | | |  | | --- | | **property\_id** | | |  | | --- | | **property\_id** | | |  | | --- | | **property\_id** | |
| 1 | 123 Elm Street | 250000.00 | 1 |
| 2 | 456 Oak Avenue | 300000.00 | 2 |
| 3 | 789 Pine Road | 200000.00 | 3 |

Transactions

Represents a completed sale or rental transaction between a client and a property.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type | Constraints |
| Transactions | transaction\_id | Unique ID for each transaction | INTEGER | Primary Key (PK) |
| client\_id | Linked client | INT | Foreign Key (FK) |
|  | property\_id | Linked property | INT | Foreign Key (FK) |
|  | date | Transaction date | DATE | CHECK date >= '2024-07-01' |
|  | amount | Transaction amount | DECIMAL(10,2) | NOT NULL |

Comments on table relationships

Transactions and Payments have a one-to-many relationship. Each transaction can have multiple payments associated with it, but a payment is linked to a single transaction.

Clients and Properties have a many-to-many relationship, resolved through the Transactions table.

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| transaction\_id | client\_id | property\_id | date | amount |
| 1 | 1 | 1 | 2024-07-15 | 250000.00 |
| 2 | 2 | 2 | 2024-08-01 | 300000.00 |
| 3 | 3 | 3 | 2024-09-10 | 200000.00 |

Visits

Represents a visit by a client to a property.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type | Constraints |
| Visits | visit\_id | Unique ID for each visit | INTEGER | Primary Key (PK) |
| client\_id | Linked client | INT | Foreign Key (FK) |
|  | property\_id | Linked property | INT | Foreign Key (FK) |
|  | visit\_date | Date of the visit | DATE | NOT NULL |

Comments on table relationships

Clients and Properties also have a many-to-many relationship through the Visits junction table. A client may visit multiple properties, and a property may have multiple visits from different clients.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| visit\_id | client\_id | property\_id | visit\_date |
| 1 | 1 | 1 | 2024-07-10 |
| 2 | 2 | 2 | 2024-07-20 |
| 3 | 3 | 3 | 2024-07-25 |

Payments

Represents a payment made by a client for a transaction.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type | Constraints |
| Payments | payment\_id | Unique ID for each payment | INTEGER | Primary Key (PK) |
| transaction\_id | Linked transaction | INT | Foreign Key (FK) |
|  | payment\_date | Date of payment | DATE | NOT NULL |
|  | amount\_paid | Amount paid | DECIMAL(10,2) | NOT NULL |

Comments on table relationships

Transactions and Payments have a one-to-many relationship. Each transaction can have multiple payments associated with it, but a payment is linked to a single transaction.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| payment\_id | transaction\_id | payment\_date | amount\_paid |
| 1 | 1 | 2024-07-20 | 125000.00 |
| 2 | 1 | 2024-07-30 | 125000.00 |
| 3 | 2 | 2024-08-10 | 150000.00 |