Refresh Constructions

“Global Redevelopment of Archaic/Old Buildings”

INFX 543: A: Relational Database Management Systems

Team Number: 5

Atinderpal Singh Saini

[atinder@uw.edu](mailto:atinder@uw.edu)

**Database Purpose:**

The purpose of this database management system is to maintain the data used to track and report on past and existing construction projects and find possible opportunities for redevelopment of an existing property to the employees of Refresh Constructions. The database would be used by administrative, marketing and sale’s teams of Refresh Constructions.

**Addressed Business Problems:**

* Allow users to identify available opportunities for redevelopment of an existing property.
* To report on the current market value of a specific property.
* To track the past economic value of an existing property.
* To identify sales patterns and trends of multiple neighborhoods.
* Supply ownership and background data necessary to lock the deal.
* Supply insight for targeted initiatives, such as: data on redevelopment of properties in smaller towns, etc.

**Business Rules:**

* Each building has a Name, Address, Neighborhoods, Architectural style, Zone, Current Condition.
* Each building will have its Sales History along with Ownership Details.
* A building may have been sold one or more times.
* Any owner may have purchased one or more buildings.

**Database Design Requirements:**

* Use Crow’s Foot Notation.
* Specify the primary key fields in each table by specifying PK beside the fields.
* Draw a line between the fields of each table to show the relationships between each table.
* Specify which table is on the one side of the relationship by placing a one next to the field where the line starts.
* Specify which table is on the many side of the relationship by placing a crow’s feet symbol next to the field where the line ends.

**Database Design Decisions:**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity Included** | **How it’s Related to Other Entities** |
| Building | This is the core entity that provides data about the commodity of interest- the building itself. It contains attributes like- ID, Name, Address, Architect’s ID, Architectural Style Code, Owner’s Type and Owner’s ID, Construction Date, Current Condition’s Code, Availability and Neighborhood. | Building\_ID is the primary key that identifies the entity. Condition Code (Cnd\_Code) is used to relate entity Condition with this entity using associative entity- ConditionHistory. This entity maybe in a given condition at a given time but can have multiple conditions over a period of time. Arch\_ID relates it to Architect’s ID and Architecture\_Code helps track the type of structure. Neighborhood entity has a one-to-many relationship with the building entity. OwnerHistory is a many-to-one relationship like the associative entity and core entity. SaleHistory’s relationship is a many-to-one with this entity |
| Condition | This entity has a PK: Cnd\_Code which is a Foreign Key in core entity “Building”. This entity stores data about the current condition of the building. | This entity has a one-to-many condition with the associative entity Condition History which has a many-to-one relationship with the core entity Building |
| ConditionHistory | This is an added associative entity. | It has a many-to-one relationship with both Condition and Building |
| Architecture | This entity contains information about the architectural type of the building. | This entity is related in one-to-many relationship with Building |
| Architect | This entity contains all the necessary background information about the architect of the building | It has a one-to-many relationship with building and also contains a foreign key City\_ID |
| Neighborhood | Contains information about specific locality of the building | Related in many-to-one relationship with City and one-to-many relationship with Building |
| City | Contains information about specific city in which the building is | Related in many-to-one relationship with State and one-to-many relationship with Neighborhood. Also has PK City\_ID referenced by both Owner and Architect entities as FK |
| State | Contains information about specific state in which the building is constructed | Related in many-to-one relationship with Country and one-to-many relationship with City |
| Country | Contains information about specific country in which the building is constructed | Related in one-to-many relationship with State |
| OwnerHistory | This entity is the associative entity between Owner and Building. A building may have multiple owners and owners may own multiple buildings. This entity provides the flexibility to make that happen. It contains FK Owner\_ID and two PKs: Building\_ID and Owner\_Start\_Date | This entity is related in many-to-one relationship with both Building and Owner |
| Owner | This entity contains all the necessary background information about the | It has a one-to-many relationship with OwnerHistory and SaleHistory and also contains a foreign key City\_ID creating a many-to-one relationship with City |
| SaleHistory | This entity serves as a mediator between Building and Owner providing more information about when did a specific group/owner buy a specific building | It has a many-to-one relationship with both Owner and Building entities |

*Refresh Consulting may use the attached ERD diagram to assess the scalability and flexibility of this project.*

**ERD:**

