IST 659 Database Administration & Database Management Concepts

Mark Roberts

Section 402 Tuesdays 7:00 p.m.-8:30 p.m.

Project 2 Deliverable

March 29, 2021

**Summary**

A real estate developer has grown fast and needs help wrangling its far-flung projects into a cohesive data system, as opposed to a collection of spreadsheets, paper documents, accounting documents, tabbed folders and collections of emails, etc.

The developer builds “Communities”. Communities are collections of “Homes” located in the suburbs. Communities may have the attributes “Normal” or “Adult Active”. Each Home is located on a Street and is designated by a Number that is distinct for that Street; Street is unique but Number is not (there could be a 100 Maple Drive and a 100 Meadow Lane); the combination of the two is unique and also forms the Street Address of the Home.

Homes come in an array of types based on the Number of Bedrooms, ranging from 1 to 5 Bedrooms. Normal Communities have Homes in the full range from 1 to 5 – Families are welcome! Adult Active Communities are for quasi-retirees 55+, with a restriction of no young children. Therefore, Homes in Adult Active Communities may only have 1 or 2 Bedrooms (enough for 1 guest – No Ruffians!)

Homes feature amenities such as car parking, swimming pool, access to fitness center, nearest school. Some features depend on the Community (for instance, Adult Active Communities are not near any schools by design). Homes will feature different car parking configurations as a function of number of bedrooms. 5 bedroom homes may have more than one kitchen.

Each community features a dedicated Sales Team. A Team must have at least one member or “Sales Agent” and may have more than one; also, a Sales Agent may be on more than one Sales Team and function as a Sales Agent for more than one Community. A member of the Sales Team will act as “Owner” in a Sale Transaction (as a representative of the developer). Each Sale Transaction will have a Buyer. The Sale Transaction will have information associated with it such as Asking Price, Days on Market, Offer Price, and Final Sale Price.

Stakeholders/Entity/Attribute List & Glossary

|  |  |  |
| --- | --- | --- |
| Stakeholders | Entities | Attributes |
| DEVELOPER  Creator of the Community/Homes, Original “Owner” | **COMMUNITIES**  Collection of Homes developed in tandem with one another. | Type:   * Adult Active * Normal |
| SALES TEAM  Agents employed by the Developer to Sell the Homes and act as Seller Agent in the Sale Transaction | **HOMES**  Individual units in Communities. Capable of being bought and sold. | Type   * 1, 2, 3, 4 or 5 Bedroom Units * Street * Number * Community Type * Amenities * Sold Status |
| SALES Transactions  Transfers of properties from Owner to Buyer via monetary purchase. | **Owner/Agent as Rep**  **Buyer**  **Home** | * Asking Price * Days on Market * Offer Price * Final Sale Price * Sale Date |
| OWNER  Individual in possession of a Home. |  | Name  Phone  Address |
| BUYER  Individual to whom the Home is sold. |  | Name  Phone  Address |
| AMENITIES  Series of features attributed to houses based on Community and Home Type. | Amenity List | Car Parking/# Cars  Nearest School  Swimming Pool  Additional Kitchen |

Conceptual & Logical Data Models

ERD

Diagram

Description automatically generated

E-ERD

Diagram

Description automatically generated

Raw Data Sample

Text, letter

Description automatically generated

Physical Database Design

-- Physical Database Design: Repeatable Script Drops/Creates Table According to Order of Dependency

-- Drop views

DROP VIEW IF EXISTS CommunityHomes;

GO

DROP VIEW IF EXISTS OwnersList;

GO

DROP VIEW IF EXISTS BuyersList;

GO

DROP VIEW IF EXISTS Sales;

GO

DROP VIEW IF EXISTS TotalAskingPriceByCommunity;

GO

DROP VIEW IF EXISTS HomesInventorybyBathrooms;

GO

DROP VIEW IF EXISTS TotalSales;

GO

-- Drop Table Statements

DROP TABLE IF EXISTS SaleTransaction;

GO

DROP TABLE IF EXISTS Buyers;

GO

DROP TABLE IF EXISTS Owners;

GO

DROP TABLE IF EXISTS AmenityTypeList;

GO

DROP TABLE IF EXISTS Homes;

GO

DROP TABLE IF EXISTS CommunitySalesTeam;

GO

DROP TABLE IF EXISTS Community;

GO

DROP TABLE IF EXISTS SalesTeam;

GO

DROP TABLE IF EXISTS BedroomType;

GO

DROP TABLE IF EXISTS Contact;

GO

DROP TABLE IF EXISTS AmenityProfiles;

GO

-- Create Table Statements, creating columns and constraints where indicated.

CREATE TABLE AmenityProfiles (

AmenityProfilesID int identity,

AmenityProfileName varchar(20) not null,

NumberCars varchar(5) not null,

SwimmingPool bit not null,

AdditionalKitchen bit not null

CONSTRAINT PK\_AmenityProfiles PRIMARY KEY (AmenityProfilesID)

);

GO

CREATE TABLE Contact (

ContactID int identity,

FirstName varchar(20) not null,

LastName varchar(20) not null,

EmailAddress varchar(20) not null,

PhoneNumber varchar(20)

-- Contraints

CONSTRAINT PK\_Contact PRIMARY KEY (ContactID),

CONSTRAINT U1\_Contact UNIQUE (EmailAddress)

);

GO

CREATE TABLE BedroomType (

BedroomTypeID int identity,

NumberBedrooms varchar(10) not null

CONSTRAINT PK\_BedroomType PRIMARY KEY (BedroomTypeID),

CONSTRAINT U1\_BedroomType UNIQUE (NumberBedrooms)

);

GO

CREATE TABLE SalesTeam (

SalesTeamID int identity,

SalesTeamName varchar(50) not null,

CommunityID int null,

ContactID int null

CONSTRAINT PK\_SalesTeam PRIMARY KEY (SalesTeamID),

CONSTRAINT FK1\_SalesTeam FOREIGN KEY (ContactID) REFERENCES Contact(ContactID)

);

GO

CREATE TABLE Community (

CommunityID int identity,

CommunityName varchar(20) not null,

CommunityType varchar(20) not null,

CommunityLocation varchar(20) not null,

NumberHomes int

CONSTRAINT PK\_Communities PRIMARY KEY (CommunityID)

);

GO

CREATE TABLE CommunitySalesTeam (

CommunitySalesTeamID int identity,

CommunityId int null,

SalesTeamID int null,

CONSTRAINT PK\_CommunitySalesTeam PRIMARY KEY (CommunitySalesTeamID),

CONSTRAINT FK1\_CommunitySalesTeam FOREIGN KEY (CommunityID) REFERENCES Community(CommunityID),

CONSTRAINT FK2\_CommunitySalesTeam FOREIGN KEY (SalesTeamID) REFERENCES SalesTeam(SalesTeamID),

);

GO

CREATE TABLE Homes (

HomesID int identity,

StreetNumber varchar(10) not null,

StreetName varchar(20) not null,

CommunityID int null,

BedroomTypeID int null,

AmenityProfilesID int null,

AskingPrice float not null

CONSTRAINT PK\_Homes PRIMARY KEY (HomesID),

CONSTRAINT FK1\_Homes FOREIGN KEY (CommunityID) REFERENCES Community(CommunityID),

CONSTRAINT FK2\_Homes FOREIGN KEY (BedroomTypeID) REFERENCES BedroomType(BedroomTypeID),

CONSTRAINT FK3\_Homes FOREIGN KEY (AmenityProfilesID) REFERENCES AmenityProfiles(AmenityProfilesID)

);

GO

CREATE TABLE AmenityTypeList (

AmenityTypeListID int identity,

AmenityProfilesID int not null,

HomesID int not null

CONSTRAINT PK\_AmenityTypeList PRIMARY KEY (AmenityTypeListID),

CONSTRAINT FK1\_AmenityTypeList FOREIGN KEY (AmenityProfilesID) REFERENCES AmenityProfiles(AmenityProfilesID),

CONSTRAINT FK2\_AmenityTypeList FOREIGN KEY (HomesID) REFERENCES Homes(HomesID)

);

GO

CREATE TABLE Owners (

OwnersID int identity,

ContactID int not null

CONSTRAINT PK\_Owners PRIMARY KEY (OwnersID),

CONSTRAINT FK1\_Owners FOREIGN KEY (ContactID) REFERENCES Contact(ContactID)

);

GO

CREATE TABLE Buyers (

BuyersID int identity,

ContactID int not null

CONSTRAINT PK\_Buyers PRIMARY KEY (BuyersID),

CONSTRAINT FK1\_Buyers FOREIGN KEY (ContactID) REFERENCES Contact(ContactID)

);

GO

CREATE TABLE SaleTransaction (

SaleTransactionID int identity,

FinalPrice float,

DaysOnMarket int,

SaleDate datetime,

HomesID int,

OwnersID int,

BuyersID int

CONSTRAINT PK\_SaleTransaction PRIMARY KEY (SaleTransactionID),

CONSTRAINT FK1\_SaleTransaction FOREIGN KEY (HomesID) REFERENCES Homes(HomesID),

CONSTRAINT FK2\_SaleTransaction FOREIGN KEY (OwnersID) REFERENCES Owners(OwnersID),

CONSTRAINT FK3\_SaleTransaction FOREIGN KEY (BuyersID) REFERENCES Buyers(BuyersID)

);

GO

Data Creation & Manipulation

-- Insert statements

-- Create Community records

INSERT INTO Community(CommunityName, CommunityLocation, NumberHomes, CommunityType)

VALUES ('Presidential Place', 'Quincy', 10, 'Gated')

, ('Seaview Cliff', 'Marblehead', 10, 'Adult Active')

, ('Quaker Village', 'Lincoln', 10, 'Retirement');

GO

SELECT \* FROM Community

-- Create Bedroom Type records

INSERT INTO BedroomType(NumberBedrooms)

VALUES ('2 BR')

, ('3 BR')

, ('4 BR');

GO

SELECT \* FROM BedroomType

-- Create Contact records

INSERT INTO Contact(FirstName, LastName, EmailAddress, PhoneNumber)

VALUES ('Abby', 'Adams', 'AAdams@domain.xyz', '(617) 555-1111')

, ('Dolly', 'Madison', 'DMadison@domain.xyz', '(617) 555-2222')

, ('Frank', 'Drake', 'FDrake@domain.xyz', '(617) 555-3333')

, ('Wally', 'Raleigh', 'WRaleigh@domain.xyz', '(617) 555-4444')

, ('Judy', 'Dench', 'JDench@domain.xyz', '(617) 555-5555')

, ('Bonny', 'Raitt', 'BRaitt@domain.xyz', '(617) 555-6666');

GO

SELECT \* FROM Contact

-- Create AmenityProfiles records

INSERT INTO AmenityProfiles(AmenityProfileName,NumberCars,SwimmingPool,AdditionalKitchen)

VALUES ('Basic','2',0,0)

, ('Deluxe','3',0,1)

, ('Luxury','4',1,1);

GO

SELECT \* FROM AmenityProfiles

-- Create Homes Records

INSERT INTO Homes(StreetNumber,StreetName,Homes.BedroomTypeID,Homes.AmenityProfilesID,AskingPrice)

VALUES

('1','Washington Street',1,1,400000)

, ('3','Washington Street',1,1,400000)

, ('5','Washington Street',2,2,550000)

, ('2','Madison Boulevard',2,2,550000)

, ('4','Madison Boulevard',3,3,700000)

, ('6','Madison Boulevard',3,3,700000)

, ('100','Minot Road',1,1,500000)

, ('300','Minot Road',1,1,500000)

, ('500','Minot Road',2,2,650000)

, ('200','Quarterdeck Lane',2,2,650000)

, ('400','Quarterdeck Lane',3,3,725000)

, ('400','Quarterdeck Lane',3,3,725000)

, ('50','Friendship Drive',1,1,350000)

, ('60','Friendship Drive',1,1,350000)

, ('70','Friendship Drive',2,2,450000)

, ('10','Meetinghouse Road',2,2,450000)

, ('20','Meetinghouse Road',3,3,600000)

, ('30','Meetinghouse Road',3,3,600000);

GO

SELECT \* FROM Homes

-- Create SalesTeam records

INSERT INTO SalesTeam (SalesTeamName,CommunityID,ContactID)

VALUES

('Presidential Place Team',(SELECT CommunityID FROM Community WHERE CommunityName = 'Presidential Place'),(SELECT ContactID FROM Contact WHERE LastName = 'Adams')),

('Presidential Place Team',(SELECT CommunityID FROM Community WHERE CommunityName = 'Presidential Place'),(SELECT ContactID FROM Contact WHERE LastName = 'Madison')),

('Seaview Cliff Team',(SELECT CommunityID FROM Community WHERE CommunityName = 'Seaview Cliff'),(SELECT ContactID FROM Contact WHERE LastName = 'Drake')),

('Seaview Cliff Team',(SELECT CommunityID FROM Community WHERE CommunityName = 'Seaview Cliff'),(SELECT ContactID FROM Contact WHERE LastName = 'Raleigh')),

('Quaker Village Team',(SELECT CommunityID FROM Community WHERE CommunityName = 'Presidential Place'),(SELECT ContactID FROM Contact WHERE LastName = 'Dench')),

('Quaker Village Team',(SELECT CommunityID FROM Community WHERE CommunityName = 'Seaview Cliff'),(SELECT ContactID FROM Contact WHERE LastName = 'Raitt'));

GO

SELECT \* FROM SalesTeam

-- Create CommunitySalesTeam Records

INSERT INTO

CommunitySalesTeam(CommunityID,SalesTeamID)

VALUES (1,1)

, (2,2)

, (3,3);

GO

SELECT \* FROM CommunitySalesTeam

-- Create Onwers Records

INSERT INTO

Owners(Owners.ContactID)

VALUES

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Adams')),

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Madison')),

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Drake')),

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Dench'))

GO

SELECT \* FROM Owners

-- Update Homes CommunityID through FK reference to Community table

UPDATE TGT

SET CommunityID=SRC.CommunityID

FROM Homes AS TGT

INNER JOIN Community AS SRC ON SRC.CommunityName = 'Presidential Place'

WHERE StreetName = 'Washington Street';

GO

UPDATE TGT

SET CommunityID=SRC.CommunityID

FROM Homes AS TGT

INNER JOIN Community AS SRC ON SRC.CommunityName = 'Presidential Place'

WHERE StreetName = 'Madison Boulevard';

GO

UPDATE TGT

SET CommunityID=SRC.CommunityID

FROM Homes AS TGT

INNER JOIN Community AS SRC ON SRC.CommunityName = 'Seaview Cliff'

WHERE StreetName = 'Minot Road';

GO

UPDATE TGT

SET CommunityID=SRC.CommunityID

FROM Homes AS TGT

INNER JOIN Community AS SRC ON SRC.CommunityName = 'Seaview Cliff'

WHERE StreetName = 'Quarterdeck Lane';

GO

UPDATE TGT

SET CommunityID=SRC.CommunityID

FROM Homes AS TGT

INNER JOIN Community AS SRC ON SRC.CommunityName = 'Quaker Village'

WHERE StreetName = 'Friendship Drive';

GO

UPDATE TGT

SET CommunityID=SRC.CommunityID

FROM Homes AS TGT

INNER JOIN Community AS SRC ON SRC.CommunityName = 'Quaker Village'

WHERE StreetName = 'Meetinghouse Road';

GO

SELECT \* FROM Community;

GO

SELECT \* FROM Homes;

GO

SELECT \* FROM BedroomType;

GO

UPDATE TGT

SET BedroomTypeID=BDR.BedroomTypeID

FROM Homes AS TGT

INNER JOIN BedroomType AS BDR ON BDR.NumberBedrooms = '2 BR'

WHERE StreetName = 'Washington Street' AND StreetNumber = '1';

GO

UPDATE TGT

SET BedroomTypeID=BDT.BedroomTypeID

FROM Homes AS TGT

INNER JOIN BedroomType AS BDT ON BDT.NumberBedrooms = 2

WHERE StreetName = 'Meetinghouse Road' AND StreetNumber = '3';

GO

UPDATE TGT

SET BedroomTypeID=BDT.BedroomTypeID

FROM Homes AS TGT

INNER JOIN BedroomType AS BDT ON BDT.NumberBedrooms = 3

WHERE StreetName = 'Meetinghouse Road' AND StreetNumber = '5';

GO

UPDATE TGT

SET BedroomTypeID=BDT.BedroomTypeID

FROM Homes AS TGT

INNER JOIN BedroomType AS BDT ON BDT.NumberBedrooms = 3

WHERE StreetName = 'Meetinghouse Road' AND StreetNumber = '5';

GO

SELECT \* FROM Homes

-- Create Contact to populate Buyers table

INSERT INTO Contact(FirstName, LastName, EmailAddress, PhoneNumber)

VALUES ('Clark', 'Kent', 'CKent@domain.xyz', '(617) 555-7777')

, ('Diana', 'Prince', 'DPrince@domain.xyz', '(617) 555-8888')

, ('Bruce', 'Wayne', 'BWayne@domain.xyz', '(617) 555-9999')

, ('Wally', 'West', 'WWest@domain.xyz', '(617) 555-1010');

GO

-- Add Buyers Records

INSERT INTO

Buyers(Buyers.ContactID)

VALUES

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Kent')),

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Prince')),

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'Wayne')),

((SELECT Contact.ContactID FROM Contact WHERE LastName = 'West'))

GO

-- Create SaleTransaction Records

INSERT INTO SaleTransaction(FinalPrice,DaysOnMarket,SaleDate,SaleTransaction.HomesID,SaleTransaction.OwnersID,SaleTransaction.BuyersID)

VALUES

(680000,64,GetDate(),5,1,1)

, (380000,53,GetDate(),2,2,2)

, (625000,53,GetDate(),9,3,3)

, (705000,53,GetDate(),11,4,4)

GO

SELECT \* FROM SaleTransaction

-- Create Views

-- Lists Homes showing each's community and bedroom type

CREATE VIEW dbo.CommunityHomes

AS

SELECT

CommunityName,

StreetNumber,

StreetName,

AskingPrice,

BedroomType.NumberBedrooms

FROM Homes

JOIN Community ON Community.CommunityID = Homes.CommunityID

JOIN BedroomType ON BedroomType.BedroomTypeID = Homes.BedroomTypeID;

GO

-- Lists contacts designated as Owners

CREATE VIEW dbo.OwnersList

AS

SELECT

FirstName AS OwnerFirstName,

LastName AS OwnerLastName

FROM Owners

JOIN Contact ON Contact.ContactID = Owners.ContactID;

GO

-- Lists contacts designated as Buyers

CREATE VIEW dbo.BuyersList

AS

SELECT

FirstName AS BuyerFirstName,

LastName AS BuyerLastName

FROM Buyers

JOIN Contact ON Contact.ContactID = Buyers.ContactID;

GO

-- Lists sale transactions with Asking and Final Price

CREATE VIEW dbo.Sales

AS

SELECT

StreetNumber,

StreetName,

AskingPrice,

FinalPrice

FROM SaleTransaction

JOIN Homes ON Homes.HomesID = SaleTransaction.HomesID

GO

-- Lists total value of each community by summing asking price for each one's homes

CREATE VIEW dbo.TotalAskingPricebyCommunity

AS

SELECT

CommunityName

, SUM(AskingPrice) AS AskingPriceTotal

, COUNT(HomesID) AS CountOfHomes

FROM

Homes

RIGHT JOIN Community ON Community.CommunityID = Homes.CommunityID

GROUP BY CommunityName

GO

-- Lists number of bedroom type units per community

CREATE VIEW dbo.HomesInventorybyBathrooms

AS

SELECT

CommunityName

, NumberBedrooms

, COUNT(Homes.BedroomTypeID) AS CountOfBedroomTypes

, SUM(AskingPrice) AS AskingPriceTotal

FROM

Homes

RIGHT JOIN BedroomType ON BedroomType.BedroomTypeID = Homes.BedroomTypeID

RIGHT JOIN Community ON Community.CommunityID = Homes.CommunityID

GROUP BY CommunityName

, NumberBedrooms;

GO

-- Lists sale transactions with Asking and Final Price

CREATE VIEW dbo.TotalSales

AS

SELECT

StreetNumber,

StreetName,

CommunityName,

NumberBedrooms,

AskingPrice,

FinalPrice

FROM

SaleTransaction

INNER JOIN Homes ON Homes.HomesID = SaleTransaction.HomesID

INNER JOIN BedroomType ON BedroomType.BedroomTypeID = Homes.BedroomTypeID

INNER JOIN Community ON Community.CommunityID = Homes.CommunityID

GO

Answering Data Questions

|  |
| --- |
| 1. Count of Bedroom Types by Community |
| Table  Description automatically generated |
|  |
| 1. Total Asking Price Value by Community |
| Table  Description automatically generated |
|  |
| 1. Total Sales with Asking & Final Price |
| Graphical user interface, table  Description automatically generated |
|  |
| 1. Count of Team Members By Team |
|  |
|  |
| 1. Count of Homes by Amenity Type |
| Table  Description automatically generated with medium confidence |

Data Samples, Implementation, Reflection & Summary

# Data Samples

|  |
| --- |
| SELECT \* FROM Community; |
|  |
|  |
| SELECT \* FROM AmenityProfiles; |
|  |
|  |
| SELECT \* FROM Homes; |
|  |

# Reports

|  |
| --- |
| Data Maintenance Form – Enter New Home |
| Graphical user interface  Description automatically generated |

|  |
| --- |
| Total Asking Price By Community |
|  |
|  |
| Sales with Asking & Final Price with Sum |
|  |
|  |
| Inventory of Bathroom Units by Community |
| Table  Description automatically generated |
|  |
| Inventory of Homes with Amenity Type by Community |
| Table  Description automatically generated |
|  |
| List of Sales Team Members |
| Table  Description automatically generated |

# Reflection

My biggest reflection is scale. I might have chosen a database type with fewer facets and intricacies. Given the limited time of the quarterly session, one ought to keep feasibility in mind. With that, I was only able to achieve what I consider to be rudimentary basics, with time being the essential missing ingredient. Scale is key.

I was not able to implement in a robust fashion the concept of Contacts > Buyers, Sellers and Agents. I would like to have created a triggering system utilizing Functions & Stored Procedures to insert new homes, new contacts, update number of homes in a community by referencing foreign key counts. This was a good lesson in biting off what you can actually chew.

# Summary

The process of developing business rules, and each stage of the logical model to normalization, created a fairly seamless data insertion process. I utilized Access and Excel ODBC connections to create my reports relying heavily on Views I created. In Excel, I was able to create new data using simple formulas (SUM, AVG, Basic Math) as well as formatting information which is raw int/float in SQL but needs to have a dollar sign and commas for IRL usage. I was lucky to have a fairly deep understanding of Excel and Access to complete these tasks. I would like to have used R but I simply did not have enough time to add an unknown variable into my project. That being said, as a Data Science degree student, it would have been a useful thing to have attempted.