**Table of Contents**

[1. ER Model 3](#_Toc125926084)

[2. List of Mapped Relations 4](#_Toc125926085)

[3. Data Dictionary 5](#_Toc125926086)

[4. SQL Statements for Database Creation 18](#_Toc125926087)

[5. Database Table Contents 43](#_Toc125926088)

[6. Queries 52](#_Toc125926089)

# ER Model

Diagram

Description automatically generated

Amendments:

* The “leads” relationship between the “Staff” and “Team” relations have been altered from total participation to partial participation, removing the mandatory requirement for the presence of a Team Leader within a “Team”.



# List of Mapped Relations



# Data Dictionary

**Award** relation: Stores the details of an award

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| AwardID | Unique identifier of each award | char(3) | Primary Key  First character must be ‘A’ followed by numerical values | No |
| AwardName | Name of award | varchar(100) |  | No |
| AwardAmt | Amount received from award | smallmoney |  | Yes |

**Buyer** relation: A subtype of a member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| BuyerID | Unique identifier of each buyer | char(9) | Primary Key  Foreign Key -> Member(MemberID) | No |

**Bump** relation: stores details of each bump

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null Value** |
| BumpID | Unique identifier of each bump | char(5) | Primary key  First 2 characters must be ‘BI’ followed by numerical values | No |
| BumpDesc | Description of bump purchased | varchar(100) |  | No |
| BumpPrice | Price of bump purchased | smallmoney |  | No |

**BumpOrder** relation: Stores the details of a bump order by a seller in order to promote his/her listing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| BumpOrderID | Unique identifier of each order bump | char(20) | Primary Key  First 2 characters must be ‘BO’ followed by numerical values | No |
| PurchaseDate | The date of purchase | date | PurchaseDate not later than today | No |
| SellerID | Identifies each unique member | char(9) | Foreign Key -->  Member  (SellerID) | No |
| BumpID | Identifies each unique bump | char(5) | Foreign Key -->  Bump  (BumpID) | No |
| ListingID | Identifies each unique listing | char(20) | Foreign Key -->  Listing  (ListingID) | No |

**Category** relation: The category of a listing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| CatID | Unique identifier of each category | char(3) | Primary Key  First character must be ‘C’ followed by numerical values | No |
| CatDesc | The description of a listing category | varchar(200) |  | No |

**Chat** relation: Stores details of chat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| ChatID | Unique identifier of each chat | char(15) | Primary Key  First 2 characters must be ‘CH’ followed by numerical values | No |
| BuyerID | Unique identifier of each buyer | char(9) | Foreign Key à Buyer(BuyerID) | No |
| ListingID | Unique identifier of each listing | char(20) | Foreign Key à Listing (ListingID) | No |

**ChatMsg** relation: Stores the details of each message that is part of a chat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| ChatID | Unique identifier of each chat | char(15) | Primary Key  Foreign key -->  Chat (ChatID) | No |
| MsgSN | Message sequence number | char(4) | Primary Key | No |
| MsgDateTime | Date and time of each message sent | smalldatetime | MsgDateTime not later than today | No |
| Originator | Identifies the person who sends the message | varchar(6) | Values: Seller, Buyer | No |
| Msg | Chat messages in each chat | varchar(300) |  | No |

**Feedback** relation: Stores the details of a feedback reported by a member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| FdBkID | Unique identifier of each feedback | char(10) | Primary key  First 2 characters must be ‘FB’ followed by numerical values | No |
| FbkDesc | Description of feedback | varchar(500) |  | No |
| FbkDateTime | Date and time received for feedback | smalldatetime | FbkDateTime not later than today | No |
| FbkStatus | Status of feedback | varchar(10) | Values: Pending, Receiving, Attention, Completed | No |
| SatisfactionRank | Buyer rank of satisfaction for Seller | tinyint | Values: 1, 2, 3, 4, 5 | Yes |
| FbkCatID | Unique identifier of each feedback category | char(4) | Foreign key --> FeedbkCat(FbkCatID) | No |
| OnMemberID | Unique identifier for each member being reported | char(9) | Foreign key --> Member(MemberID) | Yes |
| ByMemberID | Unique identifier of each reporter | char(9) | Foreign key --> Member(MemberID) | No |
| StaffID | Unique identifier of each staff | char(5) | Foreign key --> Staff(StaffID) | Yes |

**FeedbkCat** relation: Stores the details of a Feedback category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| FbkCatID | Unique identifier of each feedback categories | char(4) | Primary Key  First 2 characters must be ‘FC’ followed by numerical values | No |
| FbkCatDesc | Description of feedback categories | varchar(50) |  | No |

**Follower** relation: Stores details of each follower

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| MemberID | Unique identifier of each member | char(9) | Primary Key  Foreign Key -->  Member  (MemberID) | No |
| FollowerID | Unique identifier of follower | char(9) | Primary Key  Foreign Key -->  Member (MemberID) | No |
| DateStarted | The date that a member starts following another member | date | DateStarted not later than today | No |

**Likes** relation: Stores details of all likes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| BuyerID | Unique identifier of each member | char(9) | Primary Key  Foreign Key -->  Buyer(BuyerID) | No |
| ListingID | Unique identifier of each listing | char(20) | Primary Key  Foreign Key -->  Listing  (ListingID) | No |
| DateLiked | Describes the date when the like was given | date | DateLiked not later than today | No |

**Listing** relation: Stores the details of an item listed by a seller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| ListingID | Unique identifier of each listing | char(20) | Primary Key  First 2 characters must be ‘LI’ followed by numerical values | No |
| ListStatus | Status of the listing | varchar(9) | Values: Available, Sold, Inactive | No |
| ListPrice | The price set by seller for his/her listing | smallmoney | Minimum: $0.10 | No |
| ListDateTime | The time and date when listing is made | smalldatetime | ListDateTime not later than today | No |
| ListDesc | Description of the listing | varchar(200) |  | No |
| SellerID | Unique identifier of each member | char(9) | Foreign Key à  Seller (SellerID) | No |
| SubCatID | Unique identifier of each subcategories | char(4) | Foreign Key à  SubCategory (SubCatID) | Yes |

Derived Data: ListDays is derived from ListDateTime. ListDays = Current day – Day of ListDateTime

**ListingPhoto** relation: Stores details of each listing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| ListingID | Unique identifier of each listing | char(20) | Primary Key  Foreign Key -->  Listing(ListingID) | No |
| Photo | Picture of listing | varchar(8) | Primary Key | No |

**Member** relation: Stores the details of a member of Mokesell

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| MemberID | Unique identifier of each member | char(9) | Primary Key  First 2 characters must be ‘ME’ followed by numerical values | No |
| MemberEmail | Describes the email address of each member | varchar(50) |  | No |
| MemberDateJoined | Describes the date of when each member joined  Default to today’s date) | date | MemberDateJoined not later than today | No |
| MemberMobile | Describes the mobile number of each member | char(8) | Only numerical characters are allowed | No |
| City | Name of the city each member is from  (Default value: Singapore) | varchar(20) |  | No |
| MemberName | Describes the name of each member | varchar(50) |  | No |
| MemberDOB | Describes the date of birth of each member | date | At least 16 years old | No |

Assumptions:

1. Only Singaporean phone numbers are allowed at the moment, with the format 9999 9999. International phone numbers will be included into the database when MokeSell expands its services to overseas countries in the future.

**Offer** relation: Stores the details of an offer from a buyer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| OfferID | Unique identifier of each offer | char(10) | Primary Key  First 2 characters must be ‘OF’ followed by numerical values | No |
| OfferDate | Describes the date offer was made | date | OfferDate not later than today | No |
| OfferPrice | Describes the price of the offer | smallmoney | Minimum: $0.10 | No |
| OfferStatus | Describes the status of the offer | varchar(10) | Values: Submitted, Pending, Accepted, Rejected, Completed | No |
| ListingID | Unique identifier of each listing | char(20) | Foreign key --> Listing(ListingID) | No |
| BuyerID | Unique identifier of each member | char(9) | Foreign key --> Buyer(BuyerID) | No |

**Review** relation: Stores the details of a review from a seller or buyer after an offer is completed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| ReviewID | Unique identifier of each review | char(11) | Primary Key  First 2 characters must be ‘RE’ followed by numerical values | No |
| MemberType | Describes the type of member | varchar(6) | Values: Seller, Buyer | No |
| ItemRank | Describes the rank of the item | tinyint | Values: 1, 2, 3, 4, 5 | Yes |
| DeliveryRank | Describes the rank of the delivery | tinyint | Values: 1, 2, 3, 4, 5 | Yes |
| CommunicationRank | Describes the rank of communication | tinyint | Values: 1, 2, 3, 4, 5 | No |
| Comment | Comments for the review | varchar(200) |  | Yes |
| ReviewDate | Date when review was made | date | ReviewDate not later than today | No |
| OfferID | Unique identifier of each offer | char(10) | Foreign Key -->  Offer(OfferID) | No |

**Staff** relation: Stores the details of a Mokesell staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| StaffID | Unique identifier of each staff | char(5) | Primary Key  First 2 characters must be ‘SI’ followed by numerical values | No |
| StaffDateJoined | Date that staff joined the company (Default to today’s date) | date | DateJoin not later than today | No |
| StaffName | The name of a staff | varchar(20) |  | No |
| StaffMobile | Contact number of staff | char(8) | Only numbers are allowed | No |
| TeamID | Unique identifier of each team | char(3) | Foreign key -->  Team(TeamID) | No |

**Seller** relation: A subtype of a member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| SellerID | Unique identifier of each team | char(9) | Primary Key  Foreign key -->  Member(MemberID) | No |

**Subcategory** relation: The subcategory of a listing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| SubCatID | Unique identifier of subcategory | char(4) | Primary Key  First 2 characters must be ‘SC’ followed by numerical values | No |
| SubCatDesc | Description of each subcategory | varchar(50) |  | No |
| CatID | Unique identifier of each staff | char(3) | Foreign key -->  Category(CatID) | Yes |

**Team** relation: Stores the details of a staff team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| TeamID | Unique identifier of each team | char(3) | Primary Key  First character must be ‘T’ followed by numerical values | No |
| TeamName | Name of team | varchar(30) |  | No |
| TeamLeaderID | Unique identifier of each staff | char (5) | Foreign key -->  Staff(StaffID) | Yes |

**Win** relation: Stores details of each team award

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data type** | **Constraint** | **Null value** |
| TeamID | Unique identifier of each team | char(3) | Primary Key  First character must be ‘T’ followed by numerical values | No |
| AwardID | Unique identifier of each award | char(3) | Foreign key à  Award(AwardID) | No |
| DateAwarded | Date that award was given out | date | DateAwarded not later than today | No |

# SQL Statements for Database Creation

USE master;

IF EXISTS(select \* from sys.databases where name='MokeSellP10T1')

DROP DATABASE MokeSellP10T1;

GO

Create Database MokeSellP10T1;

GO

use MokeSellP10T1;

GO

/\* Table Deletion \*/

if exists (select \* from sysobjects

where id = object\_id('Award') and sysstat & 0xf = 3)

drop table Award;

GO

if exists (select \* from sysobjects

where id = object\_id('Bump') and sysstat & 0xf = 3)

drop table Bump;

GO

if exists (select \* from sysobjects

where id = object\_id('Category') and sysstat & 0xf = 3)

drop table Category;

GO

if exists (select \* from sysobjects

where id = object\_id('FeedbkCat') and sysstat & 0xf = 3)

drop table FeedbkCat;

GO

if exists (select \* from sysobjects

where id = object\_id('Member') and sysstat & 0xf = 3)

drop table Member;

GO

if exists (select \* from sysobjects

where id = object\_id('Staff') and sysstat & 0xf = 3)

drop table Staff;

GO

if exists (select \* from sysobjects

where id = object\_id('Team') and sysstat & 0xf = 3)

drop table Team;

GO

if exists (select \* from sysobjects

where id = object\_id('Buyer') and sysstat & 0xf = 3)

drop table Buyer;

GO

if exists (select \* from sysobjects

where id = object\_id('Feedback') and sysstat & 0xf = 3)

drop table Feedback;

GO

if exists (select \* from sysobjects

where id = object\_id('Follower') and sysstat & 0xf = 3)

drop table Follower;

GO

if exists (select \* from sysobjects

where id = object\_id('Subcategory') and sysstat & 0xf = 3)

drop table Subcategory;

GO

if exists (select \* from sysobjects

where id = object\_id('Seller') and sysstat & 0xf = 3)

drop table Seller;

GO

if exists (select \* from sysobjects

where id = object\_id('Win') and sysstat & 0xf = 3)

drop table Win;

GO

if exists (select \* from sysobjects

where id = object\_id('Listing') and sysstat & 0xf = 3)

drop table Listing;

GO

if exists (select \* from sysobjects

where id = object\_id('BumpOrder') and sysstat & 0xf = 3)

drop table BumpOrder;

GO

if exists (select \* from sysobjects

where id = object\_id('Chat') and sysstat & 0xf = 3)

drop table Chat;

GO

if exists (select \* from sysobjects

where id = object\_id('Likes') and sysstat & 0xf = 3)

drop table Likes;

GO

if exists (select \* from sysobjects

where id = object\_id('ListingPhoto') and sysstat & 0xf = 3)

drop table ListingPhoto;

GO

if exists (select \* from sysobjects

where id = object\_id('Offer') and sysstat & 0xf = 3)

drop table Offer;

GO

if exists (select \* from sysobjects

where id = object\_id('ChatMsg') and sysstat & 0xf = 3)

drop table ChatMsg;

GO

if exists (select \* from sysobjects

where id = object\_id('Review') and sysstat & 0xf = 3)

drop table Review;

GO

/\* Table Creation \*/

CREATE TABLE Award

(

AwardID char(3) CHECK (AwardID LIKE 'A[0-9][0-9]'),

AwardName varchar(100) NOT NULL,

AwardAmt smallmoney NULL,

CONSTRAINT PK\_Award PRIMARY KEY (AwardID)

);

CREATE TABLE Bump

(

BumpID char(5) CHECK (BumpID LIKE 'BI[0-9][0-9][0-9]'),

BumpDesc varchar(100) NOT NULL,

BumpPrice smallmoney NOT NULL

CONSTRAINT PK\_Bump PRIMARY KEY (BumpID)

);

CREATE TABLE Category

(

CatID char(3) CHECK (CatID LIKE 'C[0-9][0-9]'),

CatDesc varchar(200) NOT NULL,

CONSTRAINT PK\_Category PRIMARY KEY(CatID)

);

CREATE TABLE FeedbkCat

(

FbkCatID CHAR(4) NOT NULL CHECK (FbkCatID LIKE 'FC[0-9][0-9]'),

FbkCatDesc VARCHAR(50) NOT NULL,

CONSTRAINT PK\_FeedbkCat PRIMARY KEY (FbkCatID)

);

CREATE TABLE Member

(

MemberID char(9) CHECK (LEFT(MemberID, 2) = 'ME' AND ISNUMERIC(RIGHT(MemberID, 7)) = 1 AND MemberID NOT LIKE '% '),

MemberEmail varchar (50) NOT NULL,

MemberDateJoined date NOT NULL DEFAULT (GETDATE()),

MemberMobile char (8) NOT NULL,

City varchar (20) NOT NULL DEFAULT('Singapore'),

MemberName varchar(50) NOT NULL,

MemberDOB date NOT NULL CHECK (DATEDIFF(year, MemberDOB, GETDATE())>= 16),

CONSTRAINT PK\_Member PRIMARY KEY(MemberID),

);

CREATE TABLE Staff

(

StaffID char(5) CHECK (StaffID LIKE 'SI[0-9][0-9][0-9]'),

StaffName varchar(20) NOT NULL,

StaffDateJoined date NOT NULL,

StaffMobile char(8) NOT NULL,

TeamID char(3) NOT NULL,

Constraint PK\_Staff Primary Key(StaffID)

);

CREATE TABLE Team

(

TeamID char(3) CHECK (TeamID LIKE 'T[0-9][0-9]'),

TeamName varchar(30) NOT NULL,

TeamLeaderID char(5) NULL,

Constraint PK\_Team Primary Key(TeamID),

Constraint FK\_Team\_TeamLeaderID Foreign Key(TeamLeaderID) References Staff(StaffID)

);

ALTER TABLE Staff ADD CONSTRAINT FK\_Staff\_TeamID FOREIGN KEY (TeamID) REFERENCES Team(TeamID);

CREATE TABLE Buyer

(

BuyerID char(9),

CONSTRAINT PK\_Buyer PRIMARY KEY (BuyerID),

CONSTRAINT FK\_Buyer\_BuyerID FOREIGN KEY (BuyerID) REFERENCES Member(MemberID),

)

CREATE TABLE Feedback

(

FbkID char(10) CHECK (LEFT(FbkID, 2) = 'FB' AND ISNUMERIC(RIGHT(FbkID, 8)) = 1 AND FbkID NOT LIKE '% '),

FbkDesc varchar(500) NOT NULL,

FbkDateTime smalldatetime NOT NULL DEFAULT (GETDATE()),

FbkStatus varchar(10) NOT NULL CHECK (FbkStatus IN ('Pending', 'Receiving', 'Attention', 'Completed')),

SatisfactionRank tinyint NULL CHECK (SatisfactionRank IN (1, 2, 3, 4, 5)),

FbkCatID char(4) NOT NULL,

OnMemberID char(9) NULL,

ByMemberID char(9) NOT NULL,

StaffID char(5) NULL,

CONSTRAINT PK\_Feedback PRIMARY KEY(FbkID),

CONSTRAINT FK\_Feedback\_FbkCatID FOREIGN KEY (FbkCatID) REFERENCES FeedbkCat(FbkCatID),

CONSTRAINT FK\_Feedback\_OnMemberID FOREIGN KEY (OnMemberID) REFERENCES Member(MemberID),

CONSTRAINT FK\_Feedback\_ByMemberID FOREIGN KEY (ByMemberID) REFERENCES Member(MemberID),

CONSTRAINT FK\_Feedback\_StaffID FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),

);

CREATE TABLE Follower

(

MemberID char(9) NOT NULL,

FollowerID char(9) NOT NULL,

DateStarted date NOT NULL CHECK (DateStarted <= GETDATE()),

CONSTRAINT PK\_Follower PRIMARY KEY (MemberID, FollowerID),

CONSTRAINT FK\_Follower\_MemberID FOREIGN KEY (MemberID) REFERENCES Member(MemberID),

CONSTRAINT FK\_Follower\_FollowerID FOREIGN KEY (FollowerID) REFERENCES Member(MemberID)

);

CREATE TABLE Subcategory

(

SubCatID char(4) CHECK (SubCatID LIKE 'SC[0-9][0-9]'),

SubCatDesc varchar(50) NOT NULL,

CatID char(3) null,

Constraint PK\_Subcategory Primary Key(SubCatID),

Constraint FK\_Subcategory\_CatID Foreign Key (CatID) References Category(CatID)

);

CREATE TABLE Seller

(

SellerID char(9),

CONSTRAINT PK\_Seller PRIMARY KEY (SellerID),

CONSTRAINT FK\_Seller\_SellerID FOREIGN KEY (SellerID) REFERENCES Member(MemberID),

);

CREATE TABLE Win

(

TeamID char(3) NOT NULL,

AwardID char(3) NOT NULL,

DateAwarded date NOT NULL Check (DateAwarded <= GetDate()),

Constraint FK\_Win\_TeamID Foreign Key (TeamID) References Team(TeamID),

Constraint FK\_Win\_AwardID Foreign Key (AwardID) References Award(AwardID)

);

CREATE TABLE Listing

(

ListingID char(20) NOT NULL CHECK (LEFT(ListingID, 2) = 'LI' AND ISNUMERIC(RIGHT(ListingID, 18)) = 1 AND ListingID NOT LIKE '% '),

ListStatus varchar(9) NOT NULL CHECK (ListStatus IN ('Available', 'Sold', 'Inactive')),

ListPrice smallmoney NOT NULL CHECK (ListPrice >= 0.10),

ListDateTime smalldatetime NOT NULL CHECK (ListDateTime <= GETDATE()),

ListDesc varchar(200) NOT NULL,

SellerID char(9) NOT NULL,

SubCatID char(4) NOT NULL,

CONSTRAINT PK\_Listing PRIMARY KEY (ListingID),

CONSTRAINT FK\_Listing\_SellerID FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),

CONSTRAINT FK\_Listing\_SubCatID FOREIGN KEY (SubCatID) REFERENCES Subcategory(SubCatID)

);

CREATE TABLE BumpOrder

(

BumpOrderID char(20) CHECK (LEFT(BumpOrderID, 4) = 'BOID' AND ISNUMERIC(RIGHT(BumpOrderID, 16)) = 1 AND BumpOrderID NOT LIKE '% '),

PurchaseDate date NOT NULL CHECK (PurchaseDate <= GETDATE()),

SellerID char(9) NOT NULL,

ListingID char(20) NOT NULL,

BumpID char(5) NOT NULL,

CONSTRAINT PK\_BumpOrder PRIMARY KEY (BumpOrderID),

CONSTRAINT FK\_BumpOrder\_SellerID FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),

CONSTRAINT FK\_BumpOrder\_ListingID FOREIGN KEY (ListingID) REFERENCES Listing(ListingID),

CONSTRAINT FK\_BumpOrder\_BumpID FOREIGN KEY (BumpID) REFERENCES Bump(BumpID)

);

CREATE TABLE Chat

(

ChatID char(15) CHECK (LEFT(ChatID, 2) = 'CH' AND ISNUMERIC(RIGHT(ChatID, 13)) = 1 AND ChatID NOT LIKE '% '),

BuyerID char(9) NOT NULL,

ListingID char(20) NOT NULL,

CONSTRAINT PK\_Chat PRIMARY KEY (ChatID),

CONSTRAINT FK\_Chat\_BuyerID FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID),

CONSTRAINT FK\_Chat\_ListingID FOREIGN KEY (ListingID) REFERENCES Listing(ListingID)

);

CREATE TABLE Likes

(

BuyerID char(9) NOT NULL,

ListingID char(20) NOT NULL,

DateLiked date NOT NULL CHECK (DateLiked <= GETDATE()),

CONSTRAINT PK\_Likes PRIMARY KEY (BuyerID, ListingID),

CONSTRAINT FK\_Likes\_BuyerID FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID),

CONSTRAINT FK\_Likes\_ListingID FOREIGN KEY (ListingID) REFERENCES Listing(ListingID)

);

CREATE TABLE ListingPhoto

(

ListingID char(20),

Photo varchar(8) CHECK (Photo LIKE 'Photo[0-9][0-9][0-9]'),

CONSTRAINT PK\_ListingPhoto PRIMARY KEY(ListingID, Photo),

CONSTRAINT FK\_ListingPhoto\_ListingID FOREIGN KEY (ListingID) REFERENCES Listing(ListingID)

);

CREATE TABLE Offer

(

OfferID char(10),

OfferDate date NOT NULL DEFAULT (GETDATE()) CHECK (OfferDate <= GETDATE()),

OfferPrice smallmoney NOT NULL CHECK (OfferPrice >= 0.10),

OfferStatus varchar (10) NOT NULL CHECK (OfferStatus IN ('Submitted', 'Pending', 'Accepted', 'Rejected', 'Completed')),

ListingID char(20) NOT NULL,

BuyerID char(9) NOT NULL,

CONSTRAINT PK\_Offer PRIMARY KEY(OfferID),

CONSTRAINT FK\_Offer\_ListingID FOREIGN KEY (ListingID) REFERENCES Listing(ListingID),

CONSTRAINT FK\_Offer\_BuyerID FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID)

);

CREATE TABLE ChatMsg

(

ChatID char(15),

MsgSN char(4) CHECK (MsgSN LIKE '[0-9][0-9][0-9][0-9]'),

Msg varchar(300) NOT NULL,

Originator varchar(6) NOT NULL CHECK (Originator IN ('Seller', 'Buyer')),

MsgDateTime smalldatetime NOT NULL DEFAULT (GETDATE()),

CONSTRAINT PK\_ChatMsg PRIMARY KEY(ChatID, MsgSN),

CONSTRAINT FK\_ChatMsg\_ChatID FOREIGN KEY (ChatID) REFERENCES Chat(ChatID),

);

CREATE TABLE Review

(

ReviewID char(11) CHECK (LEFT(ReviewID, 2) = 'RE' AND ISNUMERIC(RIGHT(ReviewID, 9)) = 1 AND ReviewID NOT LIKE '% '),

MemberType varchar(6) NOT NULL CHECK (MemberType IN ('Seller','Buyer')),

ItemRank tinyint NULL CHECK (ItemRank IN (1,2,3,4,5)),

DeliveryRank tinyint NULL CHECK (DeliveryRank IN (1,2,3,4,5)),

CommunicationRank tinyint NOT NULL CHECK (CommunicationRank IN (1,2,3,4,5)),

Comment varchar(200) NULL,

ReviewDate date NOT NULL DEFAULT (GETDATE()),

OfferID char(10) NOT NULL,

CONSTRAINT PK\_Review PRIMARY KEY (ReviewID),

CONSTRAINT FK\_Offer\_OfferID FOREIGN KEY (OfferID) REFERENCES Offer(OfferID)

);

/\* Tuple Insertion \*/

Insert into Award

values

('A01', 'Tier 1 Team (Overall)', 200),

('A02', 'Tier 2 Team (Overall)', 150),

('A03', 'Tier 3 Team (Overall)', 100),

('A04', 'Most Improved Team', 30),

('A05', 'Most Proactive Team', 30),

('A06', 'Best Dressed Team', null),

('A07', 'Excellent Service', null),

('A08', 'Most Passionate Team', 30),

('A09', 'Rockstar Rookies', null),

('A10', 'Around the Clock Contributors', null);

Insert into Bump

values

('BI001', 'Bump once – instantly' ,5),

('BI002', 'Bump daily – for 3 days' ,7),

('BI003', 'Bump daily – for 4 days' ,9),

('BI004', 'Bump daily – for 5 days' ,11),

('BI005', 'Bump daily – for 6 days' ,13),

('BI006', 'Bump daily – for 7 days' ,15),

('BI007', 'Bump daily – for 8 days' ,16),

('BI008', 'Bump daily – for 9 days' ,17),

('BI009', 'Bump twice a day for 5 days ' ,20),

('BI010', 'Bump thrice a day for 5 days' ,23);

Insert into Category

values

('C01','Others'),

('C02','Women’s Fashion'),

('C03','Men’s Fashion'),

('C04','Luxury'),

('C05','Property'),

('C06','Stationary'),

('C07','Furniture'),

('C08','Babies and Kids'),

('C09','Beauty and Personal Care'),

('C10','Computers and Tech');

Insert into FeedbkCat

values

('FC01', 'Application Issues'),

('FC02', 'Suggestions and Recommendations'),

('FC03', 'User Satisfaction'),

('FC04', 'Listing Duplications'),

('FC05', 'Suspicious Account Activity'),

('FC06', 'Offensive User Behavior or Content'),

('FC07', 'Sales of Counterfeit Products'),

('FC08', 'Usage of Irrelevant Keywords'),

('FC09', 'Transaction Issues'),

('FC10', 'Sales of Prohibited Products');

Insert into Member (MemberID,MemberEmail,MemberDateJoined,MemberMobile,City,MemberName,MemberDOB)

values

('ME0000001', 'yellow@email.com','2020-06-01','99999999','Singapore','Wen','1996-02-11'),

('ME0000002', 'judogang321@email.com','2020-06-01','99999998','Singapore','Jia Yi', '1992-04-10'),

('ME0000003','amymymy@email.com','2020-06-01','99999997','Singapore','Amy', '1999-07-03'),

('ME0000004','dbest@email.com','2020-06-01','99999996','Singapore','Alice','1986-03-30'),

('ME0000005','sunleiii@email.com','2020-06-01','99999995','Singapore','Sun Lei','1994-08-14'),

('ME0000006','tjc@email.com','2020-06-01','99999994','Singapore','Charis','1997-01-01'),

('ME0000007','thg@email.com','2020-06-01','99999993','Singapore','Hock Guan','1998-08-12'),

('ME0000008','fabulousfabian@email.com','2020-06-01','99999992','Singapore','Fabian', '1987-06-11'),

('ME0000009','drdiv@email.com','2020-06-01','99999991','Singapore','Divyan','1993-03-02'),

('ME0000010','arthur123@email.com','2020-06-01','99999990','Singapore','Arthur', '2002-01-09'),

('ME0000011','barney@email.com','2020-06-01','99873842','Singapore','Bernice', '1999-01-08'),

('ME0000012','messibarcaforever@email.com','2020-06-01','92173678','Singapore','Zhi Ren', '1988-03-01'),

('ME0000013','gaybriel@email.com','2020-06-01','86666666','Singapore','Gabriel', '1977-11-08'),

('ME0000014','ahboonz@email.com','2020-06-01','99229988','Singapore','Boon Leong', '1987-06-11'),

('ME0000015','kelvinklien@email.com','2020-06-01','99333555','Singapore','Kelvin', '1993-03-02'),

('ME0000016',' investmentking@email.com','2020-06-01','89900000','Singapore','Claudio', '2002-01-09'),

('ME0000017','floorballmaster@email.com','2020-06-01','99369369','Singapore','Ernest', '1994-08-14'),

('ME0000018','peiqiosnfrep@email.com','2020-06-01','98765432','Singapore','Pei Qi','1997-01-01'),

('ME0000019',' ahbeng@email.com','2020-06-01','81237861','Singapore','Bob', '1998-12-08'),

('ME0000020',' johndoe@email.com','2020-06-01','91288398','Singapore','John','1987-06-11');

Insert into Team (TeamID, TeamName, TeamLeaderID)

values

('T01', 'Board of Directors', null),

('T02', 'Marketing', null),

('T03', 'Sales', null),

('T04', 'Logistics', null),

('T05', 'Human Resource', null),

('T06', 'Publicity', null),

('T07', 'Accounting & Finance', null),

('T08', 'Production', null),

('T09', 'IT', null),

('T10', 'Customer Service', null);

Insert into Staff (StaffID,StaffName,StaffDateJoined,StaffMobile,TeamID)

values

('SI001', 'Danielle Bolton', '2020-03-12', '99998889', 'T01'),

('SI002', 'Sara Rojas','2020-03-12', '99998888', 'T01'),

('SI003', 'Anaya Shelton', '2020-03-12', '99998887', 'T01'),

('SI004', 'Lila Estes','2020-03-12', '99998886', 'T02'),

('SI005', 'Sadie Richardson', '2020-03-12', '99998885', 'T02'),

('SI006', 'Leia Potter','2020-03-12', '99998884', 'T02'),

('SI007', 'Faiza Booker', '2020-03-12', '99998883', 'T02'),

('SI008', 'Natasha Estrada ','2020-03-12', '99998882', 'T02'),

('SI009', 'Jazmine Bloggs', '2020-03-12', '99998881', 'T02'),

('SI010', 'Ivan Velasquez', '2020-03-12', '99998880', 'T02'),

('SI011', 'Marnie Cotton ', '2020-03-12', '99998899', 'T02'),

('SI012', 'Marvin Ball ','2020-03-12', '99998898', 'T02'),

('SI013', 'Andreas Chen ','2020-03-12', '99998897', 'T03'),

('SI014', 'Aaron Hubbard', '2020-03-12', '99998896', 'T03'),

('SI015', 'Jaxson Khan', '2020-03-12', '99998895' ,'T03'),

('SI016', 'Lorenzo Parks', '2020-03-12' , '99998894', 'T03'),

('SI017', 'Chester Harvey ','2020-03-12', '99998893', 'T03'),

('SI018', 'Ashwin Lyons', '2020-03-12', '99998892', 'T03'),

('SI019', 'Ali Tapia ','2020-03-12', '99998891', 'T03'),

('SI020', 'Joshua Patel ','2020-03-12', '99998890', 'T03'),

('SI021', 'Sana Romero', '2020-03-12', '99998999', 'T03'),

('SI022', 'Haseeb Navarro ','2020-03-12', '99998998', 'T04'),

('SI023', 'Anita O’Sullivan', '2020-03-12', '99998997', 'T04'),

('SI024', 'Marshall Goodwin', '2020-03-12', '99998996', 'T04'),

('SI025', 'Homer Hopkins', '2020-03-12', '99998995', 'T04'),

('SI026', 'Garfield Shaw', '2020-03-12', '99998994', 'T04'),

('SI027', 'Santiago Blankenship ','2020-03-12', '99998993', 'T05'),

('SI028', 'Syeda Middleton', '2020-03-12', '99998992', 'T05'),

('SI029', 'Inaaya Levine ','2020-03-12', '99998991', 'T05'),

('SI030', 'Alex Cobb', '2020-03-12', '99998990', 'T05'),

('SI031', 'Chanel Collins ','2020-03-12', '88889999', 'T06'),

('SI032', 'Elodie Mosley', '2020-03-12', '88889998', 'T06'),

('SI033', 'Eric Mack', '2020-03-12', '88889997', 'T06'),

('SI034', 'Marissa Wu', '2020-03-12', '88889996', 'T07'),

('SI035', 'Dewi Riley ','2020-03-12', '88889995', 'T07'),

('SI036', 'Saarah Miles', '2020-03-12', '88889994', 'T07'),

('SI037', 'Laila Cross ','2020-03-12', '88889993', 'T08'),

('SI038', 'Elisa Cain ','2020-03-12', '88889992', 'T08'),

('SI039', 'Ayla Mahoney ', '2020-03-12', '88889991', 'T08'),

('SI040', 'Douglas Daugherty ','2020-03-12', '88889990', 'T08'),

('SI041', 'Leyla Pham ','2020-03-12', '66666669', 'T09'),

('SI042', 'Mohsin Arnold', '2020-03-12', '66666668', 'T09'),

('SI043', 'Freddie Mcclure','2020-03-12', '66666667', 'T09'),

('SI044', 'Abdulrahman Bishop','2020-03-12', '66666666', 'T09'),

('SI045', 'Amanda Diaz','2020-03-12', '66666665', 'T09'),

('SI046', 'Olly Yoder','2020-03-12', '66666664', 'T09'),

('SI047', 'Noah Lucas','2020-03-12', '66666663', 'T09'),

('SI048', 'Niall Page','2020-03-12', '66666662', 'T10'),

('SI049', 'Neave Cummings','2020-03-12', '66666661', 'T10'),

('SI050', 'Juliette Mcleod','2020-03-12', '66666660', 'T10');

UPDATE Team

SET TeamLeaderID = CASE TeamID

WHEN 'T01' THEN 'SI001'

WHEN 'T02' THEN 'SI004'

WHEN 'T03' THEN 'SI013'

WHEN 'T04' THEN 'SI022'

WHEN 'T05' THEN 'SI027'

WHEN 'T06' THEN 'SI031'

WHEN 'T07' THEN 'SI034'

WHEN 'T08' THEN 'SI037'

WHEN 'T09' THEN 'SI041'

WHEN 'T10' THEN 'SI048'

ELSE NULL

END

WHERE TeamID IN('T01','T02','T03','T04','T05','T06','T07','T08','T09','T10');

Insert into Buyer

values

('ME0000002'),

('ME0000004'),

('ME0000006'),

('ME0000008'),

('ME0000010'),

('ME0000012'),

('ME0000014'),

('ME0000016'),

('ME0000018'),

('ME0000020');

Insert into Feedback

values

('FB00000001', 'Bug', '2020-09-02 07:39:01', 'Completed', 1, 'FC01', NULL, 'ME0000001', 'SI050'),

('FB00000002', 'App UI improvement', '2020-09-04 14:45:08', 'Completed', 3, 'FC02', NULL, 'ME0000004', 'SI048'),

('FB00000003', 'Slow loading', '2020-09-05 00:02:05', 'Completed', 2, 'FC01', NULL, 'ME0000005', 'SI050'),

('FB00000004', 'App crash', '2020-09-05 23:59:09', 'Completed', 2, 'FC01', NULL, 'ME0000005', 'SI048'),

('FB00000005', 'Sales of Prohibited Products', '2020-09-07 07:57:43', 'Completed', 5, 'FC10', 'ME0000007', 'ME0000008', 'SI049'),

('FB00000006', 'App crash', '2020-09-08 18:33:38', 'Completed', 4, 'FC01', NULL, 'ME0000002', 'SI048'),

('FB00000007', 'Sales of Prohibited Products', '2020-09-08 09:45:11', 'Completed', 5, 'FC01', 'ME0000010', 'ME0000009', 'SI049'),

('FB00000008', 'Poor customer service', '2020-10-09 23:23:49', 'Completed', 5, 'FC01', NULL, 'ME0000001', 'SI049'),

('FB00000009', 'Slow loading', '2020-10-10 11:58:50', 'Completed', 1, 'FC01', NULL, 'ME0000003', 'SI050'),

('FB00000010', 'Bug', '2020-10-11 13:32:43', 'Completed', 1, 'FC01', NULL, 'ME0000002', 'SI050');

Insert into Follower

values

('ME0000001', 'ME0000003', '2020-07-01'),

('ME0000001', 'ME0000006', '2020-07-03'),

('ME0000001', 'ME0000007', '2020-07-06'),

('ME0000003', 'ME0000001', '2020-07-08'),

('ME0000004', 'ME0000002', '2020-07-09'),

('ME0000004', 'ME0000005', '2020-07-09'),

('ME0000005', 'ME0000001', '2020-07-10'),

('ME0000005', 'ME0000003', '2020-07-12'),

('ME0000005', 'ME0000009', '2020-07-13'),

('ME0000007', 'ME0000009', '2020-08-01'),

('ME0000008', 'ME0000010', '2020-08-02');

Insert into Subcategory (SubcatID, SubCatDesc, CatID)

values

('SC01', 'Dress', 'C02'),

('SC02', 'Maternity Wear', 'C02'),

('SC03', 'Stockings', 'C02'),

('SC04', 'Top', 'C03'),

('SC05', 'Bottom', 'C03'),

('SC06', 'Bag', 'C04'),

('SC07', 'Watches', 'C04'),

('SC08', 'For Sale', 'C05'),

('SC09', 'Rental', 'C05'),

('SC10', 'Condominium', 'C05'),

('SC11', 'Pens', 'C06'),

('SC12', 'Pencil', 'C06'),

('SC13', 'Sofa', 'C07'),

('SC14', 'Bed', 'C07'),

('SC15', 'Kitchen Appliances', 'C07'),

('SC16', 'Diapers', 'C08'),

('SC17', 'Baby monitor', 'C08'),

('SC18', 'Face cream', 'C09'),

('SC19', 'Hands & Nails', 'C09') ,

('SC20', 'Mouse', 'C10') ,

('SC21', 'Desktop', 'C10');

Insert into Seller (SellerID)

values

('ME0000001'),

('ME0000003'),

('ME0000005'),

('ME0000007'),

('ME0000009'),

('ME0000011'),

('ME0000013'),

('ME0000015'),

('ME0000017'),

('ME0000019');

Insert into Win (TeamID, AwardID, DateAwarded)

values

('T01', 'A01', '2022-01-31'),

('T02', 'A02', '2022-01-30'),

('T03', 'A03', '2022-01-30'),

('T04', 'A04', '2022-03-19'),

('T05', 'A05', '2022-02-01'),

('T06', 'A06', '2022-08-12'),

('T07', 'A07', '2022-09-13'),

('T08', 'A08', '2022-05-24'),

('T09', 'A09', '2022-06-06'),

('T10', 'A10', '2022-06-06');

Insert into Listing (ListingID,ListStatus,ListPrice,ListDateTime,ListDesc,SellerID,SubCatID)

values

('LI000000000000000001','Sold',60.00,'2020-07-19 13:23:44','E-Shock 100m Water Resistant Digital Watch','ME0000003','SC07'),

('LI000000000000000002','Inactive',29.99,'2020-07-19 12:03:04','Foldable Lazy Sofa Couch with Removable Cover','ME0000007','SC13'),

('LI000000000000000003','Sold',15.50,'2020-07-19 19:30:02','Men’s Business Casual Slim Vest L','ME0000001','SC04'),

('LI000000000000000004','Sold',319.99,'2020-07-19 22:50:28','Premium Fabric L-Shape Sofa IDEA','ME0000009','SC13'),

('LI000000000000000005','Available',110.00,'2020-07-19 21:34:45','Legitech gaming mouse Ggez RGB','ME0000013','SC20'),

('LI000000000000000006','Inactive',105.00,'2020-07-19 18:38:32','Razor DeathMinum V2 Wireless Mouse','ME0000015','SC20'),

('LI000000000000000007','Sold',19.99,'2020-07-19 23:35:11','Pocco Pants XXXL','ME0000019','SC16'),

('LI000000000000000008','Sold',200.00,'2020-07-19 11:13:28','Pear Watch SE GPS 2020','ME0000017','SC07'),

('LI000000000000000009','Sold',56.99,'2020-07-19 12:13:14','Mayer Airfryer','ME0000011','SC15'),

('LI000000000000000010','Sold', 10.50,'2020-07-19 15:16:17','Ergonomic General Purpose Mouse','ME0000005','SC20');

Insert into BumpOrder (BumpOrderID,PurchaseDate,SellerID,ListingID,BumpID)

values

('BOID0000000000000001', '2020-07-19', 'ME0000003', 'LI000000000000000001', 'BI001'),

('BOID0000000000000002', '2020-08-09', 'ME0000007', 'LI000000000000000002', 'BI002'),

('BOID0000000000000003', '2020-08-12', 'ME0000001', 'LI000000000000000003', 'BI003'),

('BOID0000000000000004', '2020-08-24', 'ME0000009', 'LI000000000000000004', 'BI004'),

('BOID0000000000000005', '2020-08-03', 'ME0000013', 'LI000000000000000005', 'BI005'),

('BOID0000000000000006', '2020-08-01', 'ME0000015', 'LI000000000000000006', 'BI006'),

('BOID0000000000000007', '2020-08-17', 'ME0000019', 'LI000000000000000007', 'BI007'),

('BOID0000000000000008', '2020-08-01', 'ME0000017', 'LI000000000000000008', 'BI008'),

('BOID0000000000000009', '2020-08-29', 'ME0000011', 'LI000000000000000009', 'BI009'),

('BOID0000000000000010', '2020-08-12', 'ME0000005', 'LI000000000000000010', 'BI010');

Insert into Chat (ChatID, BuyerID, ListingID)

values

('CH0000000000001', 'ME0000006', 'LI000000000000000001'),

('CH0000000000002', 'ME0000008', 'LI000000000000000002'),

('CH0000000000003', 'ME0000008', 'LI000000000000000005'),

('CH0000000000004', 'ME0000008', 'LI000000000000000009'),

('CH0000000000005', 'ME0000010', 'LI000000000000000009'),

('CH0000000000006', 'ME0000012', 'LI000000000000000003'),

('CH0000000000007', 'ME0000018', 'LI000000000000000002'),

('CH0000000000008', 'ME0000014', 'LI000000000000000002'),

('CH0000000000009', 'ME0000016', 'LI000000000000000001'),

('CH0000000000010', 'ME0000020', 'LI000000000000000009'),

('CH0000000000011', 'ME0000002', 'LI000000000000000003'),

('CH0000000000012', 'ME0000006', 'LI000000000000000002'),

('CH0000000000013', 'ME0000004', 'LI000000000000000002'),

('CH0000000000014', 'ME0000018', 'LI000000000000000001'),

('CH0000000000015', 'ME0000002', 'LI000000000000000009');

Insert into Likes (BuyerID, ListingID, DateLiked)

values

('ME0000002', 'LI000000000000000002', '2020-07-29'),

('ME0000002', 'LI000000000000000004', '2020-07-29'),

('ME0000004', 'LI000000000000000002', '2020-07-30'),

('ME0000004', 'LI000000000000000003', '2020-07-30'),

('ME0000004', 'LI000000000000000005', '2020-08-11'),

('ME0000006', 'LI000000000000000003', '2020-08-11'),

('ME0000006', 'LI000000000000000009', '2020-09-04'),

('ME0000006', 'LI000000000000000010', '2020-09-29'),

('ME0000008', 'LI000000000000000008', '2020-09-29'),

('ME0000012', 'LI000000000000000004', '2020-09-29'),

('ME0000010', 'LI000000000000000004', '2020-09-30');

Insert into ListingPhoto (ListingID,Photo)

values

('LI000000000000000001','Photo001'),

('LI000000000000000002','Photo001'),

('LI000000000000000003','Photo001'),

('LI000000000000000004','Photo001'),

('LI000000000000000004','Photo002'),

('LI000000000000000005','Photo001'),

('LI000000000000000006','Photo001'),

('LI000000000000000007','Photo001'),

('LI000000000000000008','Photo001'),

('LI000000000000000009','Photo001');

Insert into Offer

values

('OF0000001', '2020-08-20', 60.00, 'Completed', 'LI000000000000000001', 'ME0000006'),

('OF0000002', '2020-10-22', 15.00, 'Pending', 'LI000000000000000002', 'ME0000008'),

('OF0000003', '2020-10-24', 20.00, 'Completed', 'LI000000000000000003', 'ME0000008'),

('OF0000004', '2020-11-25', 320.00, 'Completed', 'LI000000000000000004', 'ME0000008'),

('OF0000005', '2020-11-15', 90.00, 'Rejected', 'LI000000000000000005', 'ME0000010'),

('OF0000006', '2020-11-14', 85.00, 'Rejected', 'LI000000000000000006', 'ME0000012'),

('OF0000007', '2020-11-24', 20.00, 'Completed', 'LI000000000000000007', 'ME0000018'),

('OF0000008', '2020-11-29', 200.00, 'Completed', 'LI000000000000000008', 'ME0000014'),

('OF0000009', '2020-12-04', 57.00, 'Completed', 'LI000000000000000009', 'ME0000016'),

('OF0000010', '2020-12-29', 10.50, 'Completed', 'LI000000000000000010', 'ME0000020');

Insert into ChatMsg (ChatID,MsgSN,Msg,Originator,MsgDateTime)

values

('CH0000000000001','0001','Is this still up for sale?','Buyer','2020-08-17 13:45:00'),

('CH0000000000001','0002','Hi, yes it is','Seller','2020-08-18 14:44:00'),

('CH0000000000002','0001','Can lower the price?','Buyer','2020-10-20 09:33:00'),

('CH0000000000003','0001','Is it possible for you to deliver the item?','Buyer','2020-11-23 15:02:00'),

('CH0000000000004','0001','Is this still for sale?','Buyer','2020-12-01 16:04:00'),

('CH0000000000004','0002','Yes.','Seller','2020-12-02 22:08:00'),

('CH0000000000005','0001','Is this available?','Buyer','2020-12-03 12:09:00'),

('CH0000000000006','0001','What is the material of the vest?','Buyer','2020-10-08 15:33:00'),

('CH0000000000007','0001','Is there any other sofa for sale?','Buyer','2020-10-11 08:34:00'),

('CH0000000000008','0001','Is there any other sofa for sale?','Buyer','2020-10-14 14:48:00'),

('CH0000000000009','0001','Can the price be lower?','Buyer','2020-08-18 15:44:00'),

('CH0000000000009','0002','No, unfortunately','Seller','2020-08-18 15:49:00'),

('CH0000000000009','0003','What color options are there?','Buyer','2020-08-19 16:55:00'),

('CH0000000000010','0001','Is there a bigger version?','Buyer','2020-12-01 20:22:00'),

('CH0000000000011','0001','Can the price be lower?','Buyer','2020-10-22 15:44:00'),

('CH0000000000011','0002','No, unfortunately','Seller','2020-10-22 15:49:00'),

('CH0000000000012','0001','What color options are there?','Buyer','2020-10-20 16:55:00'),

('CH0000000000012','0002','Black, White, Red.','Seller','2020-10-20 20:22:00');

Insert into Review (ReviewID, MemberType, ItemRank, DeliveryRank, CommunicationRank, Comment, ReviewDate, OfferID)

values('RE000000001', 'Seller', NULL, NULL, 3, NULL, '2020-08-21', 'OF0000001'),

('RE000000002', 'Buyer', 5, 5, 3, 'Fantastic seller, fast and easy transaction', '2020-08-21', 'OF0000001'),

('RE000000003', 'Seller', NULL, NULL, 5, 'very good buyer, swift replies', '2020-10-26', 'OF0000003'),

('RE000000004', 'Buyer', 4, 3, 1, 'Slow response, but product was delivered on time.', '2020-10-26', 'OF0000003'),

('RE000000005', 'Buyer', 5, 5, 4, 'Great Recommended seller. very fast transaction and mailing. items arrive to me just like what i saw from the photo. thanks again', '2020-11-29', 'OF0000007'),

('RE000000006', 'Seller', NULL, NULL, 4, 'easy to deal', '2020-12-03', 'OF0000008'),

('RE000000007', 'Buyer', 3, 3, 1, 'Takes very long to reply', '2020-12-03', 'OF0000008'),

('RE000000008', 'Buyer', 5, 4, 4, 'very responsive seller, transaction quickly thanks bro cheers!!!!!!', '2020-12-05', 'OF0000009'),

('RE000000009', 'Seller', NULL, NULL, 5, NULL, '2022-12-05', 'OF0000009'),

('RE000000010', 'Buyer', 2, 1, 5, 'seller was punctual and accommodating', '2021-01-01', 'OF0000010');

/\* Display \*/

SELECT \* FROM Award;

SELECT \* FROM Bump;

SELECT \* FROM BumpOrder;

SELECT \* FROM Buyer;

SELECT \* FROM Category;

SELECT \* FROM Chat;

SELECT \* FROM ChatMsg;

SELECT \* FROM Feedback;

SELECT \* FROM FeedbkCat;

SELECT \* FROM Follower;

SELECT \* FROM Likes;

SELECT \* FROM Listing;

SELECT \* FROM ListingPhoto;

SELECT \* FROM Member;

SELECT \* FROM Offer;

SELECT \* FROM Review;

SELECT \* FROM Seller;

SELECT \* FROM Staff;

SELECT \* FROM SubCategory;

SELECT \* FROM Team;

SELECT \* FROM Win;

# Database Table Contents

Award

Table, Excel

Description automatically generated

Bump

Table

Description automatically generated

BumpOrder

Table

Description automatically generated

Buyer

Table

Description automatically generated with medium confidence

Category

Table

Description automatically generated

Chat

Table

Description automatically generated

ChatMsg

Table

Description automatically generated

Feedback

Graphical user interface, application

Description automatically generated

FeedbkCat

Table

Description automatically generated

Follower

Table

Description automatically generated

Likes

Table

Description automatically generated

Listing

A picture containing calendar

Description automatically generated

ListingPhoto

Table

Description automatically generated

Member

Table

Description automatically generated

Offer

Table

Description automatically generated

Review

Graphical user interface, application

Description automatically generated

Seller

Table

Description automatically generated with medium confidence

Staff

Table

Description automatically generated

Table, Excel

Description automatically generated

SubCategory

Table

Description automatically generated

Team

Table, Excel

Description automatically generated

Win

Table

Description automatically generated

# Queries

**Bernice:**

1. What are the names, ages and cities of customers who complete offers and which subcategories and categories do they buy from?

User: Staff from Marketing department / CEO

Purpose: This query shows the customer demographics of Mokesell’s completed offers and helps the staff from the marketing department to better understand their target group of customers and tailor their marketing efforts, products, and services to cater to the preferences of specific groups.

By understanding customer demographics, the CEO can also identify potential new markets to expand into and make more informed decisions.

This ultimately means that understanding customer demographics helps MokeSell better engage with customers, increasing chances of customer retention and outreach.

Input:

SELECT MemberName, City, DATEDIFF(YEAR, MemberDOB, GETDATE()) AS 'Age', SubCatDesc, CatDesc

FROM Member m

INNER JOIN Offer o

ON m.MemberID = o.BuyerID

INNER JOIN Listing l

ON o.ListingID = l.ListingID

INNER JOIN Subcategory sc

ON l.SubCatID = sc.SubCatID

INNER JOIN Category c

ON sc.CatID = c.CatID

GROUP BY MemberName, City, DATEDIFF(YEAR, MemberDOB, GETDATE()), sc.SubCatDesc, c.CatDesc

ORDER BY 'Age' ASC;

Output:

Table

Description automatically generated

1. Based on the completed offers in 2020, which categories and subcategories were those listings from and how much were they sold for?

User: Staff from the Marketing / IT team

Purpose: This query lets staff from the Marketing / Sales team see which categories the completed offers are under, showing the demand for the various categories for the year of 2020. By monitoring the demand and sales performance of a certain category, the staff can identify trends in customer behavior and demand. This can help them make informed decisions on product offerings and promotions that they should recommend to buyers. Furthermore, the IT staff can keep a better lookout on popular categories to minimize fraudulent transactions, ensuring that MokeSell is a reliable e-commerce platform for members.

Input:

SELECT sc.CatID,c.CatDesc AS 'Category Description',sc.SubCatID,sc.SubCatDesc AS 'Sub-Category Description', ListDesc AS 'Listing Description', ListPrice, OfferStatus, OfferDate

FROM Listing l

INNER JOIN Offer o

ON l.ListingID = o.ListingID

INNER JOIN Subcategory sc

ON l.SubcatID = sc.SubcatID

INNER JOIN Category c

ON sc.CatID = c.CatID

WHERE OfferDate BETWEEN '2020-01-01' AND '2020-12-31' AND OfferStatus = 'Completed'

ORDER BY CatID ASC;

Output:

A picture containing graphical user interface

Description automatically generated

1. What are the names of the leaders from the top 3 teams (in alphabetical order)?

User: CEO

Purpose: Talent Retention is crucial for ensuring that the company has the talent it needs to meet its business objectives and remain competitive in the marketplace.

Input:

/\* List the names of the team leaders from the top 3 teams in ascending order \*/

SELECT StaffID, StaffName

FROM Staff

WHERE StaffID IN

(SELECT TeamLeaderId

From Team

WHERE TeamID IN

(SELECT TeamID

FROM Win

WHERE AwardID IN

(SELECT AwardID

FROM Award

WHERE AwardAmt > 30)))

ORDER BY StaffName ASC;

Output:

Table

Description automatically generated

**Zhi Ren:**

1. What type of complaints have been made the most? (and which staff received the feedback)

User: Customer service / IT department

Purpose: When the app is released for public use, there may be unforeseen problems that users may run into when using the app. With this table, Customer service can find out what kind of problem users encounter the most when using the app. They would then send these results to the IT department where they can resolve the problems within the app, keeping the app up and running for users to use.

Input:

Select Count(f.FbkCatID) as 'Number of feedback made', f.FbkDesc ,s.StaffID from

Staff s inner join Feedback f on s.StaffID = f.StaffID

group by FbkDesc,s.StaffID

Output:

Graphical user interface, table

Description automatically generated

1. How often are bumps purchased?

User: Board of Directors / Publicity

Purpose: Bumps are another way the company brings in revenue and if nobody buys bumps, the company will not earn as much money as they possibly could. With this table, the Board of Directors can see how much they earn from bumps as well as the frequency of sellers buying bumps. The dates from the table indicate that sellers have not been buying bumps very often. With this information the Board of Directors can ultimately decide to try to encourage users to buy more bumps by implementing more incentives for users to buy or by asking the Publicity department to make advertisements for bumps.

Input:

Select l.SellerID ,Count(BumpOrderID) as 'Bumps purchased' ,b.BumpPrice, bo.PurchaseDate from

Listing l inner join BumpOrder bo on l.SellerID = bo.SellerID

inner join Bump b on b.BumpID = bo.BumpID group by l.SellerID,

b.BumpPrice,bo.PurchaseDate Order by BumpPrice DESC

Output:

Table

Description automatically generated

1. What awards have each team won?

User: Board of Directors / the rest of the teams

Purpose: In order to reward teams that have done well, awards will be given out. This table displays the awards each team has won, this can allow the Board of Directors to determine which team deserves a raise / reward. This can also encourage the other teams to work harder and strive for new awards, which in turn can make the company more productive.

Input:

Select t.TeamName, a.AwardName, Count(w.TeamID) as 'Number of Awards won (so far)'

from Team t inner join Win w on t.TeamID = w.TeamID inner join Award a on

a.AwardID = w.AwardID group by AwardName,TeamName

Output:

Table

Description automatically generated