CIS 3400 - ETRA

Baruch College

**ZQZD LLP**

**Investment Fund & Customers**

Alex Lin, Lishan Gao, Xianglin Chen, Shulian Liu, Yanfei Wu

Leader - Jiping.Lin@baruchmail.cuny.edu

Developer - Lishan.Gao@baruchmail.cuny.edu

Developer - Xianglin.Chen@baruchmail.cuny.edu

Documentation Writer - Shulian.Liu@baruchmail.cuny.edu

System Analyst - Yanfei.Wu@baruchmail.cuny.edu

Contents

[I. Business Scenario 3](#_Toc72246498)

[II. ER Model using UML Notation 4](#_Toc72246499)

[III. Converting the ERD into RDM 5](#_Toc72246500)

[IV. Normalization 6](#_Toc72246501)

[V. Creating Tables with SQL 9](#_Toc72246502)

[VI. Scenarios 21](#_Toc72246503)

[VII. Database Application 23](#_Toc72246504)

[VIII. Conclusion 27](#_Toc72246505)

I. Business Scenario

Our company is a small investment firm based in New York City. Our firm has been owned and operated by us for around six years. Our company does not have a database that keeps track of everything, and this has caused multiple problems in the past few years. Manual checking might lead to human error from time to time and should not be allowed in an investment environment like us, as it would cause losses not only to us but also to our customers. This year the company has decided to implement a database to minimize the possibility of human error. Additionally, we can collect investors' information to form an extensive database to identify what kind of fund each group of people prefer to buy. It would help us determine future product ideas and attract prospective customers to invest in our firm's products.

Our business offers two types of funds to be invested by our customers: the Everlast Fund and the Initiator Fund, and both of these will be. The Everlast Fund focuses on low-risk investments such as physical assets and bonds. The Initiator Fund focuses on high-risk investments such as cryptocurrencies. Our company will also offer consultants that our customers will make appointments to confer before deciding to invest in a fund. In this process, to improve our services' accuracy, we need to collect the necessary data from customers and our company—for instance, kinds of funds, personal backgrounds, and assigned consultants. We will track the liquid assets, buy information of the fund(time, kind, price), and the investor's final return report for a better marketing model. We can use these data to draw customer personas so that we can design a better selling plan. As a result, our company will provide better profitability and higher work efficiency to the public buyers.

In addition to this information, we also need to track our employees' ID, address, contact info and pay. We need to know the amount of money currently invested and the return on investment of it for our funds. Besides, we will have to track the appointment ID, the date, and the time for our appointments.

II. ER Model using UML Notation

Diagram

Description automatically generated

One **Customer** can have one or many **Customer Accounts.**

One **Customer** account can only be owned by one **Customer**.

One **Customer** can make zero or many **Appointments**.

 One **Appointment** can only be made by one **Customer**.

One **Consultant** can have zero or many **Appointments**.

One **Appointment** can only be made with one **Consultant**.

One **Consultant** can manage one or many **Customer Accounts**.

 One **Customer Accounts** can be managed by one or many **Consultant**.

 One **Payment** can pay for one or many **Bills**.

One **Bill** can be paid by one or many **Payments**.

One **Customer Accounts** can create one or many **Bills**.

One **Bill** can be created by one **Customer Account**.

One **Customer Accounts** can place one or many **Trade**.

One **Trade** can only be placed by one **Customer.**

One **Trade** involves only one **Instrument**.

One **Instrument** can be involved in one or many **Trade**.

III. Converting the ERD into RDM

This step is to convert the Entity Relationship diagram to a Relational Model. We underlined and made the note for the primary key and the foriengn key.

**Payment** (PaymentID,PaymentDate,AmountPaid,PaymentMethod)

**Bills** (BillD, bill\_amount, bill\_date, DueDate, AccountNumber) Payment\_Bills(PaymentID (fk), BillD(fk))

**Bills\_Payment**(PaymentID,BillD)

**CustomerAccount** (AccountNumber, CashBalance, TotalValue, InvestmentValue, ReturnValue, RountingNum, AccountNum,CustomerID)

**Customer**(CustomerID, FirstName, LastName,Street, City, State, Zipcode, PhoneNumber, EmailAddress, UserName, LIquidAssets, SSN)

**Appointment**(AppointmentID, App\_Date, App\_Time, CustomerID, ConsultantID)

**Consultants**(ConsultantID, FirstName, LastName, Street, City, State, Zipcode, PaySalary, Title, PhoneNumber, EmailAddress, hours\_worked,AccountNumber)

**Consultants\_CustomerAccount**(ConsultantID(fk), AccountNumber(fk))

**Trade** (TradeID, Shares, Price, trade\_amount, trade\_date, AccountNumber, Action, Symbol) Instruments(Symbol, type)

IV. Normalization

This step is to convert the Entity Relationship diagram to a Relational Model. We underlined and made the note for the primary key and the foriengn key.

**Payment** (PaymentID,PaymentDate,AmountPaid,PaymentMethod)

**Bills** (BillD, bill\_amount, bill\_date, DueDate, AccountNumber) Payment\_Bills(PaymentID (fk), BillD(fk))

**Bills\_Payment**(PaymentID,BillD)

**CustomerAccount** (AccountNumber, CashBalance, TotalValue, InvestmentValue, ReturnValue, RountingNum, AccountNum,CustomerID)

**Customer**(CustomerID, FirstName, LastName,Street, City, State, Zipcode, PhoneNumber, EmailAddress, UserName, LIquidAssets, SSN)

**Appointment**(AppointmentID, App\_Date, App\_Time, CustomerID, ConsultantID)

**Consultants**(ConsultantID, FirstName, LastName, Street, City, State, Zipcode, PaySalary, Title, PhoneNumber, EmailAddress, hours\_worked,AccountNumber)

**Consultants\_CustomerAccount**(ConsultantID(fk), AccountNumber(fk))

**Trade** (TradeID, Shares, Price, trade\_amount, trade\_date, AccountNumber, Action, Symbol) Instruments(Symbol, type)

**IV. Normalization**

In this step, we analyzed each table if there’s any Partial Dependency and Transitive Dependency by using the Relational Model. We made and processed new relations for our model.

**Payment Relation**

Payment(PaymentID,PaymentDate,AmountPaid,PaymentMethod)

Step 1: Primary Key:  PaymentID

Step 2: Functional dependency:

FD1: PaymentID → PaymentDate, AmountPaid, PaymentMethod

Step 3:

1. Having Key? Yes

2. Partial Dependency? No

3. Transitive Dependency? No

Payment(PaymentID, PaymentDate, AmountPaid, PaymentMethod)

**Bills Relation**

Bills(BillD, bill\_amount, bill\_date, DueDate, AccountNumber)

Step 1: Primary Key: BillID

Step 2: Functional dependency:

FD1: BillID → bill\_amount, bill\_data, DueDate, AccountNumber

FD2: AccountNumber → bill\_amount, bill\_data, DueDate

Step 3:

1. Having Key?Yes, FD1

2. Partial Dependency? No

3. Transitive Dependency? Yes

R1(AccountNumber, bill\_amount, bill\_data, DueDate)

R2(BillD, AccountNumber)

**Customer Account Relation**

CustomerAccount(AccountNumber, CashBalance, TotalValue, InvestmentValue, ReturnValue, RoutingNum, AccountNum)

Step 1: Primary Key: AccountNumber

Step 2: Functional dependency:

FD1: AccountNumber → CashBalance, TotalValue, InvestmentValue, ReturnValue, RoutingNum, AccountNum

Step 3:

1. Having Key? Yes

2. Partial Dependency? No

3. Transitive Dependency? No

CustomerAccount(AccountNumber, CashBalance, TotalValue, InvestmentValue, ReturnValue, RoutingNum, AccountNum)

**Customer Relation**

Customer(CustomerID, FirstName, LastName, Street, City, State, Zipcode, PhoneNumber, EmailAddress, UserName, LiquidAssets, SSN)

Step 1: Primary Key: CustomerID

Step 2: Functional dependency:

FD1: CustomerID → FirstName, LastName, Street, City, State, Zipcode, PhoneNumber, EmailAddress, UserName, LiquidAssets, SSN

FD2:  SSN →  FirstName, LastName

FD3: UserName → PhoneNumber, EmailAddress, FirstName, LastName, Street, City, State, Zipcode

FD4: Zipcode → City, State

Step 3:

1. Having Key? Yes

2. Partial Dependency? No

3. Transitive Dependency? Yes

R1(Zipcode, City, State)

R2(UserName, PhoneNumber, EmailAddress, FirstName, LastName, Street, Zipcode)

R3(SSN, FirstName, LastName)

R4(CustomerID, UserName, LiquidAssets, SSN)

**Appointment Relation**

Appointment(AppointmentID, App\_Date, App\_Time, CustomerID, ConsultantID)

Step 1: Primary Key: AppointmentID

Step 2: Functional dependency:

FD1: AppointmentID → App\_Date, App\_Time, CustomerID, ConsultantID

FD2: CustomerID → App\_Date, App\_Time

FD3: ConsultantID → App\_Date, App\_Time

Step 3:

1. Having Key? Yes, in 1NF

2. Partial Dependency? No.

3. Transitive Dependency? Yes, FD2 and FD3.

R1(CustomerID, App\_Date, App\_Time)

R2(ConsultantID, App\_Date, App\_Time)

R3(AppointmentID, CustomerID, ConsultantID)

**Consultants relation**

Consultants(ConsultantID, FirstName, LastName, Street, City, State, Zipcode, PaySalary, Title, PhoneNumber, EmailAddress, hours\_worked,AccountNumber)

Step 1: Primary Key: ConsultantID

Step 2: Functional dependency:

FD1: Consultant ID -> FirstName, LastName, Street, City, State, Zipcode, PaySalary, Title, PhoneNumber, EmailAddress, hours\_worked, AccountNumber

FD2: AccountNumber --> FirstName, LastName, PhoneNumber, EmailAddress

FD3: Zipcode-->City, State

Step 3:

1. Having Key? Yes

2. Partial Dependency? No

3. Transitive Dependency? Yes

R1(Zipcode, City, State)

R2(AccountNumber, FirstName, LastName, PhoneNumber, EmailAddress)

R3(ConsultantID, Street, Zipcode, PaySalary, Title, AccountNumber, hours\_worked)

**Trade Relation**

Trade(TradeID, Shares, Price, trade\_amount, trade\_date, AccountNumber, Action,  Symbol)

Step 1: Primary Key:TradeID

Step 2: Functional dependency:

FD1:TradeID → Shares, Price, trade\_amount, trade\_date, AccountNumber, Action, Symbol

FD2:Symbol → Price

FD3: AccountNumber → Shares, trade\_amount, trade\_date, Action

Step 3:

1. Having Key?Yes,FD1

2. Partial Dependency?No

3. Transitive Dependency?

R1(Symbol, Price)

R2(AccountNumber, Shares, trade\_amount, trade\_date, Action)

R2(TradeID, Shares, AccountNumber, Symbol)

**Instruments Relation**

Instruments(Symbol, type)

Step 1: Primary Key: Symbol

Step 2: Functional dependency:

FD1: Symbol → type

Step 3:

1. Having Key? Yes

2. Partial Dependency? No

3. Transitive Dependency? No

Instruments\_Details(Symbol, type)

V. Creating Tables with SQL

We created the table in the database based on the model we have. Then, we added the primary key and foriegn key to set up the relations. For “Many-to-Many” relationships, we created a new table combining the foreign keys from both sides.

**Part One: Creating Table**

CREATE TABLE Customer(

CustomerID VARCHAR(30) NOT NULL,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(20) NOT NULL,

Street VARCHAR(20),

City VARCHAR(20),

State VARCHAR(20),

ZipCode Number,

PhoneNumber VARCHAR(20),

EmailAddress VARCHAR(30),

UserName VARCHAR(20),

LiquidAssets Number,

SSN VARCHAR(11),

CONSTRAINT pk\_Customer PRIMARY KEY (CustomerID)

)

CREATE TABLE CustomerAccount (

AccountNumber VARCHAR(30) NOT NULL,

CashBalance NUMBER,

TotalValue NUMBER,

InvestmentValue NUMBER,

ReturnValue NUMBER,

RoutingNum VARCHAR(20) NOT NULL,

AccountNum VARCHAR(20) NOT NULL,

CustomerID VARCHAR(30),

CONSTRAINT pk\_CustomerAccount PRIMARY KEY(AccountNumber),

CONSTRAINT fk\_CustomerAccount FOREIGN KEY(CustomerID) REFERENCES Customer(CustomerID)

)

CREATE TABLE Payment (

PaymentID VARCHAR(30) NOT NULL,

PaymentDate DATE NOT NULL,

AmountPaid NUMBER,

PaymentMethod VARCHAR(30),

CONSTRAINT pk\_Payment PRIMARY KEY  (PaymentID)

);

CREATE TABLE Bills(

BillID VARCHAR(30) NOT NULL,

BillAmount VARCHAR(30) NOT NULL,

BillDate DATE NOT NULL,

Due\_Date DATE NOT NULL,

AccountNumber VARCHAR(20),

CONSTRAINT pk\_Bills PRIMARY KEY (BillID),

CONSTRAINT fk\_Bills FOREIGN KEY(AccountNumber) REFERENCES CustomerAccount (AccountNumber)

);

CREATE TABLE Bills\_Payment(

BillID VARCHAR(30) NOT NULL,

PaymentID VARCHAR(30) NOT NULL,

Constraint pk\_bp Primary Key(BillID,PaymentID),

Constraint fk\_bp Foreign Key(BillID) references Bills(BillID),

Constraint fk\_bp1 Foreign Key(PaymentID) references Payment(PaymentID)

);

CREATE TABLE Consultants(

ConsultantID  VARCHAR(6) NOT NULL,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(20) NOT NULL,

Street VARCHAR(40),

City VARCHAR(20),

State VARCHAR(2),

ZipCode VARCHAR(10),

PaySalary NUMBER,

Title VARCHAR(20),

PhoneNumber VARCHAR(20),

EmailAddress VARCHAR(30),

Hours\_worked NUMBER,

CONSTRAINT pk\_Consultants PRIMARY KEY(ConsultantID)

);

CREATE TABLE CustomerAccount\_Consultants(

AccountNumber VARCHAR(30),

ConsultantID VARCHAR(6),

CONSTRAINT pk\_cc Primary Key(AccountNumber, ConsultantID),

CONSTRAINT fk\_cc Foreign Key(AccountNumber) REFERENCES CustomerAccount(AccountNumber),

CONSTRAINT fk\_cc1 Foreign Key(ConsultantID) REFERENCES Consultants(ConsultantID)

);

CREATE TABLE Appointment (

AppointmentID VARCHAR(30) NOT NULL,

App\_Date DATE NOT NULL,

App\_Time VARCHAR(10) NOT NULL,

CustomerID VARCHAR(30),

ConsultantID VARCHAR(6),

CONSTRAINT pk\_Appointment PRIMARY KEY(AppointmentID),

CONSTRAINT fk\_Appointment1  FOREIGN KEY(CustomerID) REFERENCES Customer (CustomerID),

CONSTRAINT fk\_Appointment2  FOREIGN KEY(ConsultantID) REFERENCES Consultants (ConsultantID)

);

CREATE TABLE Instruments(

Symbol VARCHAR(30) NOT NULL,

Type VARCHAR(30)NOT NULL,

CONSTRAINT pk\_Instruments PRIMARY KEY(Symbol)

);

CREATE TABLE Trade (

TradeID VARCHAR(20),

Shares INT,

Price NUMBER,

Trade\_Amount NUMBER,

Trade\_Date DATE,

Action VARCHAR(30),

AccountNumber VARCHAR(30),

Symbol VARCHAR(30),

CONSTRAINT pk\_Trade PRIMARY KEY(TradeID),

CONSTRAINT fk\_Trade1 FOREIGN KEY(AccountNumber) REFERENCES CustomerAccount(AccountNumber),

CONSTRAINT fk\_Trade2 FOREIGN KEY(Symbol) REFERENCES Instruments(Symbol)

);

**Database Schema and the Relationship Model**

Graphical user interface, diagram, application, PowerPoint

Description automatically generated

**Part Two: Adding the data into the table using SQL INSERT**

**Customer**

INSERT INTO Customer VALUES ('00001', 'Amy', 'Fawcett', '5630 Van Doren St', 'Queens', 'NY', '11350', '9178309231', 'Amy01@gmail.com', 'Amy01', 1320413, '123-45-6789')

INSERT INTO Customer VALUES ('00002', 'Adele', 'Garcia', '1115 Bay Ridge Ave.', 'Brooklyn', 'NY', '11219', '6467589832', 'a.garcia@outlook.com', 'adelethesinger', 9213023, '112-33-8765')

INSERT INTO Customer VALUES ('00003', 'Basia', 'Morse', '415 73rd St', 'Brooklyn', 'NY', '11209', '7189027741', 'morbasia23@gmail.com', 'morbasia423', 942984, '298-89-1132')

INSERT INTO Customer VALUES ('00004', 'Bella', 'Smith', '782 Tompkins Ave.', 'Staten Island', 'NY', '10305', '9294242331', 'bellpepper@gmail.com', 'peppersmith22', 504209, '520-13-1400')

INSERT INTO Customer VALUES ('00005', 'Candice', 'Brown', '146 Brewster Rd', 'Scarsdale', 'NY', '10583', '6463329821', 'candicedf@yahoo.com', 'CBrown', 3293091, '091-83-9929')

INSERT INTO Customer VALUES ('00006',  'Alan', 'Miller','21 Morgan Ave', 'Norwalk', 'CT', '06851', '4759832244', 'milleralan131@gmail.com', 'amiller131', 102092, '983-77-3321')

INSERT INTO Customer VALUES ('00007', 'Chloe', 'Rodriguez', '362 Southport St', 'Ronkonkoma', 'NY', '11779', '6318729834', 'chloechloechloe@gmail.com', 'chloe', 475209, '742-09-0032')

INSERT INTO Customer VALUES ('00008', 'Dale','Davis',  '85-14 Wareham Pl', 'Jamaica', 'NY', '11432', '3479890123', 'davidsinsteadofd@hotmail.com', 'Davidwiths', 13742921, '832-91-8290')

INSERT INTO Customer VALUES ('00009', 'Ben','Lopez', '137-78 Westgate St', 'Jamaica', 'NY', '11413', '9178886529', 'benlopez210@gmail.com', 'ben10lopez', 93292, '675-20-9274')

INSERT INTO Customer VALUES ('00010', 'Yun','Zhang',  '1029 49TH Ave', 'Long Island City', 'NY', '11101', '7186356775', 'Yunzhang@gmail.com', 'Yunzhang', 44294, '345-678-8910')

INSERT INTO Customer VALUES ('00011', 'Ella', 'Lopez', '223 Clover Pl', 'Mine Hill', 'NJ', '07805', '9175890275', 'EllaLopez@gmail.com', 'Elllla11', 821024, '678-46-6710')

INSERT INTO Customer VALUES ('00012', 'Elsa', 'Davis', '998 Van Siclen Ave', 'Brooklyn', 'NY', '11207', '3479820932', 'notfrozenelsa@gmail.com', 'elsathequeen', 4989921, '892-22-8842')

INSERT INTO Customer VALUES ('00013', 'Dan', 'Taylor', '1673 Prospect Pl', 'Brooklyn', 'NY', '11233', '6468309231', 'dantaylor1673@gmail.com', 'dantaylor', 588923, '323-46-9829')

INSERT INTO Customer VALUES ('00014', 'Gill', 'Adele', '1577 Lurting Ave', 'Bronx', 'NY', '10461', '9178779800', 'GillAdele@gmail.com', 'GillA', 239912, '567-45-6889')

INSERT INTO Customer VALUES ('00015', 'Gina', 'Lee', '219 E 5th St', 'New York', 'NY', '10003', '9178331112', 'GinLee@gmail.com', 'GinL', 999520, '132-54-9823')

INSERT INTO Customer VALUES ('00016', 'Grace', 'Feng', '7655 192nd St', 'Queens', 'NY', '11366', '6464329231', 'GraceFeng01@gmail.com', 'GraceFeng', 131400, '556-41-5677')

INSERT INTO Customer VALUES ('00017', 'Zhuanqian', 'Huang',  '27 Division St', 'New York', 'NY', '10002', '3475300231', 'zhuanqian33@gmail.com', 'zqian', 4982312, '642-45-9099')

INSERT INTO Customer VALUES ('00018', 'Facai', 'Ke', '140 Anderson Ave', 'Fairview', 'NJ', '07022', '7185432990', ' Facai@gmail.com', 'Facai', 778834, '788-09-6752')

INSERT INTO Customer VALUES ('00019', 'Hulda', 'Yong', '763 57th St', 'Brooklyn', 'NY', '11220', '6462309231', 'hualda.yong@gmail.com', 'yhualda520', 918234, '091-82-5584')

INSERT INTO Customer VALUES ('00020', 'Iris', 'Baker', '617 Anderson Ave', 'Fairview', 'NJ', '07010', '9175855990', 'IrisBaker@gmail.com', 'BakerIris', 70231, '678-45-6009')

Table

Description automatically generated

(sample)

**Consultants**

INSERT INTO Consultants VALUES ('01234', 'Ana', 'Yang', '134 Driggs Street', 'Staten Island',  'NY', '10308', 5720.00, 'Junior', '646-645-3182', 'Ana.Yang@zqzd.org', 160)

INSERT INTO Consultants VALUES ('01235', 'Janice', 'Zhu', '1425 71st St', 'Brooklyn', 'NY', '11228', 9300.00, 'Senior', '718-325-3232', 'Janice.Zhu@zqzd.org', 180)

INSERT INTO Consultants VALUES ('01236', 'Kevin', 'Paredes', '914 57th St', 'Brooklyn', 'NY', '11219', 5530.21, 'Junior', '646-775-3093', 'Kevin.Paredes@zqzd.org', 154)

INSERT INTO Consultants VALUES ('01237', 'Joanna', 'Green', '1458 45th St', 'Brooklyn', 'NY', '11219', 5682.49 , 'Junior', '646-215-3399', 'Joanna.Green@zqzd.org', 158)

INSERT INTO Consultants VALUES ('01238', 'Hu', 'Ge', '1624 44th St', 'Brooklyn', 'NY', '11204', 5021.41 , 'Junior', '718-215-3119', 'Hu.Ge@zqzd.org', 145)

INSERT INTO Consultants VALUES ('01239', 'Amanda', 'Chen', '353 W 51st St', 'New York', 'NY', '10019', 9813.87 , 'Senior', '917-498-3091', 'Amanda.Chen@zqzd.org', 192)

INSERT INTO Consultants VALUES ('01240', 'Alex', 'Liang', '1107 5th Ave', 'New York', 'NY', '10128', 13923.42, 'Senior', '646-331-4745', 'Alex.Liang@zqzd.org', 201)

INSERT INTO Consultants VALUES ('01241', 'Lisa', 'West', '82 164th Pl', 'Jamaica', 'NY', '11432', 6100.20, 'Junior', '917-288-3880', 'Lisa.West@zqzd.org', 165)

INSERT INTO Consultants VALUES ('01242', 'Peter','Simth', '108 Greenway North', 'Forest Hills', 'NY', '11375', 5804.39, 'Junior', '918-390-3320', 'Peter.Smith@zqzd.org', 156)

INSERT INTO Consultants VALUES ('01243', 'Kristi', 'Lin', '142-27 Barclay Ave', 'Flushing', 'NY', '11355', 5739.33, 'Junior', '347-426-9953', 'Kristi.Lin@zqzd.org', 159)

Table

Description automatically generated(sample)

**Appointment**

INSERT INTO Appointment VALUES ('1001', '05/01/2021', '2:00 pm','00001','01237')

INSERT INTO Appointment VALUES ('1002', '05/01/2021', '3:00 pm','00005','01242')

INSERT INTO Appointment VALUES ('1003', '05/02/2021', '10:30 am','00007','01241')

INSERT INTO Appointment VALUES ('1004', '05/03/2021', '10:00 am','00004','01236')

INSERT INTO Appointment VALUES ('1005', '05/04/2021', '10:00 am','00008','01237')

INSERT INTO Appointment VALUES ('1006', '05/04/2021', '12:30 pm','00009','01238')

INSERT INTO Appointment VALUES ('1007', '05/04/2021', '13:30 pm','00011','01239')

INSERT INTO Appointment VALUES ('1008', '05/05/2021', '14:00 pm','00018','01243')

INSERT INTO Appointment VALUES ('1009', '05/06/2021', '14:00 pm','00015','01234')

INSERT INTO Appointment VALUES ('1010', '05/06/2021', '16:00 pm','00019','01235')

A picture containing graphical user interface

Description automatically generated(sample)

**CustomerAccount**

INSERT INTO CustomerAccount VALUES('918234567', 3010.20, 23020.38, 5103.40, 3150.3, '021000322','012457564','00001')

INSERT INTO CustomerAccount VALUES('918328637', 5020.68, 3120.55, 3251.33, 1032.59, '21000089','342457564','00002')

INSERT INTO CustomerAccount VALUES('918394675', 4312.12, 1463.12, 4232.00, 4061.25, '026012881','001268511','00003')

INSERT INTO CustomerAccount VALUES('918332564', 41423.56, 6542.22, 5363.00, 4322.33, '011103093','0351467041','00004')

INSERT INTO CustomerAccount VALUES('918498566', 4321.52, 4233.11, 2632.42, 1233.13, '021000322','012451168','00005')

INSERT INTO CustomerAccount VALUES('918656432', 57464.12, 14562.12, 4672.12, '3232.23', '026012881','001268511','00006')

INSERT INTO CustomerAccount VALUES('918354354', 1523.23, 2356.23, 2311.25, 1239.30, '21000089','341268911','00007')

INSERT INTO CustomerAccount VALUES('918265421', 5646.56, 12365.26, 3000.00, 2346.20, '226070584','341268911','00008')

INSERT INTO CustomerAccount VALUES('918465476', 3423.20, 12500.13, 9231.21, 3345.12, '021000322','512564505','00009')

INSERT INTO CustomerAccount VALUES('918357665', 4165.20, 8431.03, 4230.12, 4231.00, '011103093','035156709','00010')

INSERT INTO CustomerAccount VALUES('918654327', 9414.14, 15630.12, 6743.30, 3564.80, '21000089','345268811','00011')

INSERT INTO CustomerAccount VALUES('918342757', 4637.12, 9042.12, 4323.18, 2473.77, '226070584','015111564','00012')

INSERT INTO CustomerAccount VALUES('918764577', 3273.01,34642.03, 12674.11, 5423.00, '021000021','470070415','00013')

INSERT INTO CustomerAccount VALUES('918372438', 3241.45, 9873.12, 3242.12, 4574.12, '021516321','470070415','00014')

INSERT INTO CustomerAccount VALUES('918635437', 2341.23,23413.12, 12342.13, 4002.10, '021000322','012659564','00015')

INSERT INTO CustomerAccount VALUES('918917311', 3241.23, 32137.00, 12370.41, 4165.01, '026012881','350852634','00016')

INSERT INTO CustomerAccount VALUES('918626170', 9413.12,8432.03, 3140.00, 1041.11, '011103093','035856706','00017')

INSERT INTO CustomerAccount VALUES('918795101', 8713.02, 30437.03, 11264.21, 6400.10, '021000322','014575618','00018')

INSERT INTO CustomerAccount VALUES('918661470', 6342.02,5461.60, 3176.10, 1645.10, '21000089','345268252','00019')

INSERT INTO CustomerAccount VALUES('918642143', 2314.00, 324314.00, 9030.00, 4313.00, '021000021','001245745','00020')

Graphical user interface, table

Description automatically generated(sample)

**Instruments**

INSERT INTO Instruments VALUES('CRM', 'Stock ')

INSERT INTO Instruments VALUES('BA', 'Stock ')

INSERT INTO Instruments VALUES('AMZN', 'Stock ')

INSERT INTO Instruments VALUES('AMD', 'Stock ')

INSERT INTO Instruments VALUES('U ', 'Stock ')

INSERT INTO Instruments VALUES('SQ', 'Stock ')

INSERT INTO Instruments VALUES('DIS', 'Stock ')

INSERT INTO Instruments VALUES('ABNB', 'Stock ')

INSERT INTO Instruments VALUES('HD', 'Stock ')

INSERT INTO Instruments VALUES('CAT', 'Stock ')

INSERT INTO Instruments VALUES('NKE', 'Stock ')

INSERT INTO Instruments VALUES('AAPL', 'Stock ')

INSERT INTO Instruments VALUES('COST', 'Stock ')

INSERT INTO Instruments VALUES('MRNA', 'Stock ')

INSERT INTO Instruments VALUES('VTIAX', 'ETF ')

INSERT INTO Instruments VALUES('LQD', 'ETF ')

INSERT INTO Instruments VALUES('VCIT', 'ETF ')

INSERT INTO Instruments VALUES('BND', 'ETF ')

INSERT INTO Instruments VALUES('SPY', 'ETF')

INSERT INTO Instruments VALUES('QQQ', 'ETF ')

A picture containing table

Description automatically generated (sample)

**Trade**

INSERT INTO Trade VALUES('135336', 1000, 5011.35, 3000.25, '06/30/2019', 'Buy', '918234567', 'CRM')

INSERT INTO Trade VALUES('135363', 1000, 7123.55, 3231.20, '07/30/2020', 'Sell', '918234567', 'BA')

INSERT INTO Trade VALUES('135456', 2500, 5011.35, 3651.20, '04/05/2018', 'Buy', '918394675', 'AMZN')

INSERT INTO Trade VALUES('135697', 3000, 9536.55, 8321.00, '01/15/2020', 'Buy', '918332564', 'AMD')

INSERT INTO Trade VALUES('135963', 2600, 5569.43, 4323.20, '05/20/2019', 'Buy', '918498566', 'U')

INSERT INTO Trade VALUES('135631', 600, 1235.66, 982.00, '02/09/2019', 'Sell', '918656432', 'DIS')

INSERT INTO Trade VALUES('135646', 8000, 10236.89, 9861.00, '03/20/2018', 'Sell', '918354354', 'SQ')

INSERT INTO Trade VALUES('135333', 10000, 35621.10, 25698.33, '12/16/2020', 'Sell', '918265421', 'ABNB')

INSERT INTO Trade VALUES('135985', 6000, 9573.2, 8231.00, '05/10/2019', 'Buy', '918465476', 'HD')

INSERT INTO Trade VALUES('135785', 1300, 3422.33, 2130.20, '11/18/2020', 'Sell', '918465476', 'CAT')

INSERT INTO Trade VALUES('135563', 6500, 9823.23, 7253.00, '02/16/2020', 'Sell', '918357665', 'NKE')

INSERT INTO Trade VALUES('135952', 5200, 8632.43, 7102.00, '11/05/2020', 'Sell', '918654327', 'AAPL')

INSERT INTO Trade VALUES('135489', 1600, 2456.46, 1256.02, '06/17/2018', 'Buy', '918342757', 'COST')

INSERT INTO Trade VALUES('135999', 25000, 59634.60, 35621.00, '07/20/2018', 'Buy', '918764577', 'MRNA')

INSERT INTO Trade VALUES('135845', 1600, 3264.12, 2354.10, '11/18/2020', 'Sell', '918372438', 'VTIAX')

INSERT INTO Trade VALUES('135721', 2600, 9431.23, 8752.33, '06/15/2020', 'Sell', '918635437', 'LQD')

INSERT INTO Trade VALUES('135323', 2300, 3563.23, 2456.00, '02/19/2020', 'Buy', '918917311', 'VCIT')

INSERT INTO Trade VALUES('135496', 2300, 4563.00, 3201.00, '01/01/2019', 'Buy', '918626170', 'BND')

INSERT INTO Trade VALUES('135653',4500, 9564.06, 7256.20, '11/09/2019', 'Buy', '918626170', 'SPY')

INSERT INTO Trade VALUES('135500', 13100, 23246.06, 12354.03, '12/04/2018', 'Sell', '918795101', 'QQQ')

INSERT INTO Trade VALUES('135700', 4200, 8520.23, 7253.00, '03/29/2018', 'Sell', '918661470', 'BND')

Table

Description automatically generated

(sample)

**Payment**

INSERT INTO Payment VALUES('P54321','06/18/2018', 34906.54,'Cash')

INSERT INTO Payment VALUES('P54324','06/18/2018', 2512.00, 'Check')

INSERT INTO Payment VALUES('P54325','06/18/2018', 4503.63, 'Credit Card')

INSERT INTO Payment VALUES('P34341','01/22/2017', 2030.45, 'Debit Card')

INSERT INTO Payment VALUES('P34368','03/02/2019', 346.34, 'Check')

INSERT INTO Payment VALUES('P34369','03/02/2019', 3120.45, 'Credit Card')

INSERT INTO Payment VALUES('P59237','02/14/2019', 1034.94, 'Check')

INSERT INTO Payment VALUES('P61354', '01/28/2018', 3650.66, 'Cash')

INSERT INTO Payment VALUES('P61369','07/30/2018', 1120.19, 'Cash')

INSERT INTO Payment VALUES('P71369','07/30/2018', 1123.12, 'Cash')

Table

Description automatically generated(sample)

**Bills**

INSERT INTO Bills VALUES('B1122', 34906.54, '02/10/2018', '07/19/2018', '918654327')

INSERT INTO Bills VALUES('B1123', 3120.45, '06/25/2017', '07/25/2018', '918332564')

INSERT INTO Bills VALUES('B1124', 4503.63, '03/20/2018', '04/20/2018', '918234567')

INSERT INTO Bills VALUES('B1125', 5120.19, '01/15/2017', '02/15/2017', '918265421')

INSERT INTO Bills VALUES('B1126', 1034.94, '12/02/2018', '01/02/2019', '918642143')

INSERT INTO Bills VALUES('B1127', 3652.66, '10/04/2016', '11/20/2016', '918332564')

INSERT INTO Bills VALUES('B1128', 2030.45, '03/20/2018', '04/20/2018', '918328637')

INSERT INTO Bills VALUES('B1129', 34906.54, '02/21/2017', '02/21/2020', '918372438')

INSERT INTO Bills VALUES('B1130', 8436.34, '09/13/2017', '10/13/2019', '918394675')

INSERT INTO Bills VALUES('B1131', 34326.91, '12/19/2017', '01/09/2018', '918654327')

Text

Description automatically generated with medium confidence(sample)

**Bills\_Payment**

INSERT INTO Bills\_Payment VALUES('B1122', 'P54321')

INSERT INTO Bills\_Payment VALUES('B1123', 'P54324')

INSERT INTO Bills\_Payment VALUES('B1124', 'P54325')

INSERT INTO Bills\_Payment VALUES('B1125', 'P34341')

INSERT INTO Bills\_Payment VALUES('B1126', 'P34368')

INSERT INTO Bills\_Payment VALUES('B1127', 'P34369')

INSERT INTO Bills\_Payment VALUES('B1128', 'P59237')

INSERT INTO Bills\_Payment VALUES('B1129', 'P61354')

INSERT INTO Bills\_Payment VALUES('B1130', 'P61369')

INSERT INTO Bills\_Payment VALUES('B1131', 'P71369')

**Table

Description automatically generated**(sample)

**CustomerAccount\_Consultants**

INSERT INTO CustomerAccount\_Consultants VALUES ( '918234567 ','01234')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918328637','01235')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918394675','01236')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918332564','01237')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918498566','01238')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918656432 ','01239')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918354354 ','01240')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918265421','01241')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918465476 ','01242')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918357665','01243')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918654327 ','01234')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918342757','01235')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918764577','01236')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918372438','01237')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918635437','01238')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918917311 ','01239')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918626170 ','01240')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918795101','01241')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918661470','01242')

INSERT INTO CustomerAccount\_Consultants VALUES ( '918642143','01243')

A picture containing timeline

Description automatically generated(sample)

VI. Scenarios

The following examples will be a demonstration of the use of our databsase. We have three queries. Each of them presents a different condition for the company to do certain researches.

**Example 1: Consultant Evaluation**

Our investment firm wants to know who (consultants) help clients earn more money.

Find each customer’s return percentage from investment, which means the use of return value divided by investment value. And rank those percentages in descending order. The table should include his or her Customer ID, name, account number, return value, consultants’ information (ConslutantID, FirstName, LastName), and appointment information (Appointment date and appointment time).

SELECT cs.ConsultantID, cs.FirstName, cs.LastName, ct.CustomerID, ct.FirstName, ct.LastName,ca.AccountNumber,a.App\_Date, a.App\_Time, round(ca.ReturnValue/ca.InvestmentValue\*100,2) AS Return\_Percentage

FROM CustomerAccount ca,Customer ct,Appointment a,Consultants cs

WHERE ca.CustomerID=ct.CustomerID AND a.CustomerID=ct.CustomerID AND cs.ConsultantID=a.ConsultantID

ORDER BY round(ca.ReturnValue/ca.InvestmentValue\*100,2) DESC

Graphical user interface, application, table

Description automatically generated

**Example 2: Unpaid Bills Notification**

Our investment firm wants to know which customer owes bills until today so that we can send them a notification to the customer through email, mail or phone call.

The requirement will be selecting the customerid, customer name, phone number, and customer address where the total value of bills for a customer is higher than the total value of payment until today.

SELECT c.CustomerID,c.LastName, c.FirstName, c.Street, c.City,c.State,c.Zipcode, c.EmailAddress, c.PhoneNumber

FROM (SELECT CA.CustomerID, CA.AccountNumber,b.BillID, p.PaymentID, Sum(b.BillAmount) as TotalBill, Sum(p.AmountPaid) as TotalPayment

FROM CustomerAccount CA, Bills b, Payment p,Bills\_Payment bp

WHERE CA.AccountNumber = b.AccountNumber and b.BillID=bp.BillID and bp.PaymentID=p.PaymentID

GROUP BY CA.CustomerID, CA.AccountNumber,b.BillID,p.PaymentID) a, Customer c

WHERE a.CustomerID=c.CustomerID and a.TotalBill>a.TotalPayment

Graphical user interface, text, application, table

Description automatically generated

**Example 3: Payment Method**

Our investment firm would like to know the bills records if using cards as payment method.  Also, we would like to know if the bill was paid by a credit card or debit card.

The requirement will be selecting the PaymentID, BillID and Payment method and listing the data with the BillID in descending order.

SELECT p.PaymentMethod, bp.PaymentID, bp.BillID

FROM Bills\_Payment bp, payment p

WHERE p.PaymentID = bp.PaymentID AND p.PaymentMethod LIKE '\*Card\*'

ORDER BY bp.BillID DESC

Graphical user interface, application, table

Description automatically generated

VII. Database Application

The following will be the screenshots for the Navigation forms in the Access Database.

**Main Menu for the Investment Hedge Fund**

(In this report, we only include three screen cuts for the tables)

**Appointment**

**Graphical user interface, table

Description automatically generated**

Graphical user interface, application, table

Description automatically generated

**Bills**

**Graphical user interface, application, table

Description automatically generatedGraphical user interface, application, table

Description automatically generated**

**Bills\_Payment**

Graphical user interface, application

Description automatically generated

Graphical user interface, table

Description automatically generated

**Queries**

**Consultant Evaluation:**

**Graphical user interface, application

Description automatically generated**

Graphical user interface, application, table, Excel

Description automatically generated

**Unpaid Bills Notification:**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application, table, Excel

Description automatically generated**

**Payment Method:**

**Graphical user interface, application

Description automatically generated**

Graphical user interface, application, table

Description automatically generated

VIII. Conclusion

We developed a database application for an investment firm beginning with constructing an Entity Relations Model and Relationship diagram, converting normalization, setting the real database structure, and then applying cases in the real business world. Our database includes the basic format for the investment firm to achieve the goal of better customer information management and improved market selling plans, However, the amount of required information for the real investment firm should be more. For example, the database needs to expand the details in the tables such as some descriptions or PE ratios in the instruments. Moreover, we can add the return rate and customer feedback so that the consultants can make more efficient choices for the customer to make decisions on trading.

As a group we did well on most of the sections. There were definitely hiccups along the way, like how we were confused on how to create a table in SQL for tables that have many-to-many relationships to other tables. However, in the end, we were able to overcome it and establish a solid database for the company. And by using our database to its fullest potential and expanding on it as time progresses, all of our proposed benefits had and could have come true. With the application of the database, we were able to decrease the amount of human errors presented to us before this. With further implementation of the database, we can achieve our goal of using it to promote the marketing aspect of the company, as the database is still too young to be fully reliable to be used in that way.