Oracle Database Design and Implementation

This project is to develop and implement an Oracle database. See the scenario of the Investment database.

This simplified Investment Portfolio case study maintains information regarding clients that invest in stocks and mutual funds. Each client has a unique taxpayer identification number. A client may invest in stocks or mutual funds.

A stock is uniquely identified by its ticker, which is symbol on the stock exchange. The name of issue for a stock is the name of the company. A stock has a rating, which is an assessment of the stock. The principal business of a stock denotes the primary business from which the company obtains its revenue. There are several prices associated with a stock. Besides the current price of a stock, the high and low price ranges are recorded for the calendar year. The return value for a stock, which is the price appreciation, is recorded for the prior year and five-year period.

A mutual fund is also uniquely identified by a ticker symbol and has a fund name. The principal objective of the fund is also recorded (e.g. ‘G’ for growth, ‘I’ for income, and ‘S’ for stability). Similar to stocks, mutual funds record a current offering price and a high and low price range. A percent yield from investment income is also maintained. A mutual fund may be associated with at most one fund family.

A fund family has a unique identification number and the company’s name and address. A fund family, as its name implies, can have many associated mutual funds, but must have at least one mutual fund.

Based on the high level introduction to the Investment Portfolio enterprise, assume the following:

The characteristics of a client include a unique taxpayer identification number, a name, and an address. A client can invest in many stocks in many mutual funds.

The characteristics of a stock include the stock’s unique ticker on the stock exchange and additional descriptive properties that include the name of issue, a rating, the principal business, a high and low price range, the current price, and return values for the prior year and 5-year period.

The characteristics of a mutual fund include the mutual fund’s unique ticker and additional descriptive properties that include the name of the fund, its principal objective, the high and low price range, the current offering price, and the fund’s yield. A mutual fund may be associated with the most one fund family.

The characteristics of a fund family include a unique identification, a name, and a address. A fund family may have mutual funds and must be associated with at least one mutual fund.

# 1. Extract Entity Types and Identify Relationship Matrix

Based on the given Investment Portfolio Scenario in Section 3.1, four external entity types have been extracted. They are identified as: Client, Stock, Mutual Fund and Family Fund

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CLIENT | STOCK | MUTUAL FUND | FAMILY FUND |
| CLIENT | -- | INVEST | INVEST | -- |
| STOCK | INVEST | -- | -- |  |
| MUTUAL FUND | INVEST | -- | -- | ASSOCIATE |
| FAMILY FUND | -- | -- | ASSOCIATE | -- |

**Generated Business rules** The three pairs of relationships are further defined as the database business rules as follows:

One client may invest in one or more than one stock.

A stock may be bought by one or more than one client.

One client may invest in one or more than one mutual fund.

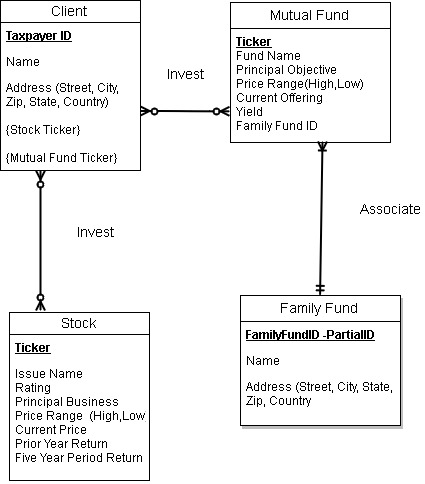
A mutual fund may be bought by one or more than one client.

One fund family must have one or more than one mutual fund.

One mutual fund must belong to only one fund family.

3. Draw an ER diagram with

1) entity types, 2) relationship types, 3) keys, 4) cardinality and 5) PKs and FKs



**Database Logical Design**

1. Map the ER diagram to a relational database schema indicating the relation name, primary key and foreign key. Add appropriate additional attribute by yourself. **NOTE: Underlined FK identifier uses referential integrity but still functions as PK in the relation**

**Client**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TaxpayerID PK | FirstName | LastName | Street | City | State | Zip | Country |

**Mutual Fund**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker PK | FundName | PrincipalObjective | PriceRangeHigh | PriceRangeLow | Current Offering | Yield | FamilyFundID FK |

**Stock**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker PK | IssueName | Rating | PrincipalBusiness | PriceRangeHigh | PriceRangeLow | CurrentPrice | PriorYearReturn | FiveYearPeriodReturn |

**Family Fund**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FamilyFundID PK | MutualFundTIcker FK | Name | Street | City | State | Zip | Country |  |

**Client Stock**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TaxpayerID FK | StockTicker FK | Quantity |  |  |

**Client Mutual Fund**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TaxpayerID FK | MutalFundTicker FK | Quantity |  |  |

1. Establish join paths for the above relational database using the referential integrity by drawing arrow lines between the above tables. Indicate all the foreign keys (FK). **NOTE: Regardless of PK Function, all inheritance denotes FK status**

Mutual Fund/FamilyFundID FK -> Family Fund/FamilyFundID PK

Client Stock/ TaxpayerID FK -> Client/TaxpayerID PK

Client Stock/ StockerTicker FK ->Stock/Ticker PK

Client Mutual Fund/ TaxpayerID FK -> Client/TaxypayerID PK

Client Mutual Fund/MutualFindTicker FK -> Mutual Fund/ Ticker PK

Family Fund/MutualFundTIcker FK -> MutualFund/ Ticker PK

1. Do function analysis for each of your tables

**NOTE: Partial D – PK determines NonKey**

**Transitive D- NonKey determines NonKey**

**Full D- PK determines all NonKey**

Attribute A -> Attribute B (Determinant attribute(s) Determines Dependent Attribute(s))

Partial D TaxpayerID -> FirstName, LastName, Street, City, State, Zip, Country

Partial D MutualFundTicker -> FundName, PrincipalObjective, PriceRangeHigh, PriceRangeLow, CurrentOffering, Yield

Partial D StockTicker -> IssueName, Rating, PrincipalBusiness, PriceRangeHigh, PriceRangeLow, CurrentPrice, PriorYearReturn, FiveYearPeriodReturn

Partial D FamilyFundIDNumber -> Name, Street, City, State, Zip, Country

Full D TaxpayerID/StockTicker -> Quantity

Full D TaxpayerID/MutualFundTicker ->Quantity

Transitive D Rating -> PrincipalBusiness, PriceRangeHigh, PriceRangeLow, CurrentPrice, PriorYearReturn, FiveYearPeriodReturn

Transitive D Yield -> CurrentOffering

1. Show all the normalized tables and indicate the normalization form for each of your tables.

**NOTE: 1NF Free of multivalued attributes + PK**

**2NF Free of Partial D + 1NF**

**3NF Free of Transitive D +2NF**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | 1NF | 2NF | 3NF |
| Client | X | X | X |
| Mutual Fund | X | X |  |
| Stock | X | X |  |
| Family Fund | X | X | X |
| Client Stock | X | X | X |
| Client Mutual Fund | X | X | X |
|  |  |  |  |

**The project would typically go through the following phases**: **DDL, DML, DCL, TCL**

**The Implemented code is as follows:**

**--CREATING CLIENT TABLE**

CREATE TABLE client\_table (

TaxpayerID INT NOT NULL,

FirstName VARCHAR (255) NOT NULL,

LastName VARCHAR (255) NOT NULL,

Street VARCHAR (255) NOT NULL,

City VARCHAR (255) NOT NULL,

State VARCHAR (255) NOT NULL,

Zip INT NOT NULL,

Country VARCHAR (255) NOT NULL,

CONSTRAINT TaxpayerID\_pk PRIMARY KEY (TaxpayerID)

);

**----INSERTING VALUES IN CLIENT TABLE**

INSERT INTO client\_table (

TaxpayerID,

Firstname ,

Lastname ,

Street ,

City ,

State ,

Zip ,

Country)

VALUES

(545786625,

'Henry',

'King',

'1234 Fake Street',

'Los Angeles',

'CA',

91754,

'USA'

);

INSERT INTO client\_table (

TaxpayerID,

Firstname ,

Lastname ,

Street ,

City ,

State ,

Zip ,

Country)

VALUES

(123456789,

'Geroge',

'Hill',

'1324 No Street',

'Los Angeles',

'CA',

91753,

'USA'

);

INSERT INTO client\_table (

TaxpayerID,

Firstname ,

Lastname ,

Street ,

City ,

State ,

Zip ,

Country)

VALUES

(987654321,

'King',

'Pin',

'1673 Narrow Street',

'Miami',

'FL',

12345,

'USA'

);

INSERT INTO client\_table (

TaxpayerID,

Firstname ,

Lastname ,

Street ,

City ,

State ,

Zip ,

Country)

VALUES

(128765930,

'Bob',

'Marley',

'666 High Blvd',

'Compton',

'CA',

66660,

'USA'

);

INSERT INTO client\_table (

TaxpayerID,

Firstname ,

Lastname ,

Street ,

City ,

State ,

Zip ,

Country)

VALUES

(943270000,

'Nick',

'Sanders',

'123 Green Blvd',

'Seatlle',

'WA',

88765,

'USA'

);

INSERT INTO client\_table (

TaxpayerID,

Firstname ,

Lastname ,

Street ,

City ,

State ,

Zip ,

Country)

VALUES

(987765401,

'Donald',

'Bautista',

'556 Tall Ave',

'Tampa',

'FL',

98987,

'USA'

);

**--CREATING STOCK TABLE**

CREATE TABLE stock\_table (

StockID VARCHAR(255) NOT NULL,

IssueName VARCHAR (255) NOT NULL,

Rating VARCHAR (100) NOT NULL,

PrincipalBuiness VARCHAR (255) NOT NULL,

PriceRangeHigh Decimal(38,2) NOT NULL,

PrinceRangeLow Decimal(38,2) NOT NULL,

CurrentPrice Decimal(38,2) NOT NULL,

PriorYearReturn Decimal(38,2),

FiveYearPeriodReturn Decimal(38,2),

CONSTRAINT StockID\_pk PRIMARY KEY (StockID)

);

**--INSERTING VALUES IN STOCK TABLE**

INSERT INTO stock\_table

VALUES

('12345',

'ABC',

'7',

'Retail',

123,

100,

102,

109,

108

);

INSERT INTO stock\_table

VALUES

('67890',

'MNO',

'4',

'Wholeseller',

787,

710,

778,

750,

767

);

INSERT INTO stock\_table

VALUES

('11234',

'PQR',

'6',

'Restraunt',

500,

400,

450,

455,

480

);

INSERT INTO stock\_table

VALUES

('56789',

'STU',

'2',

'Stockist',

900,

700,

800,

702,

710

);

INSERT INTO stock\_table

VALUES

('21345',

'XYZ',

'1',

'Cold Storage',

1100,

1050,

1090,

1080,

1093

);

**--CREATING CLIENTSTOCK TABLE**

CREATE TABLE clientstock\_table (

TaxpayerID INT NOT NULL,

StockID VARCHAR (255) NOT NULL,

Quantity VARCHAR(255) NOT NULL,

CONSTRAINT TaxpayerID\_fk FOREIGN KEY (TaxpayerID) REFERENCES client\_table (TaxPayerID) on delete cascade,

CONSTRAINT StockID\_fk FOREIGN KEY (StockID) REFERENCES stock\_table (StockId) on delete cascade );

**--INSERTING VALUES INTO CLIENTSTOCK TABLE**

INSERT INTO clientstock\_table (

TaxpayerID,

StockID,

Quantity)

VALUES

((SELECT TaxpayerID

FROM client\_table WHERE TaxpayerID = 545786625),

(SELECT StockID

FROM stock\_table WHERE StockID = 12345),

'400');

INSERT INTO clientstock\_table (

TaxpayerID,

StockID,

Quantity)

VALUES

((SELECT TaxpayerID

FROM client\_table WHERE TaxpayerID = 123456789),

(SELECT StockID

FROM stock\_table WHERE StockID = 67890),

'200');

INSERT INTO clientstock\_table (

TaxpayerID,

StockID,

Quantity)

VALUES

((SELECT TaxpayerID

FROM client\_table WHERE TaxpayerID = 987654321),

(SELECT StockID

FROM stock\_table WHERE StockID = 11234),

'350');

INSERT INTO clientstock\_table (

TaxpayerID,

StockID,

Quantity)

VALUES

((SELECT TaxpayerID

FROM client\_table WHERE TaxpayerID = 128765930),

(SELECT StockID

FROM stock\_table WHERE StockID = 56789),

'660');

INSERT INTO clientstock\_table (

TaxpayerID,

StockID,

Quantity)

VALUES

((SELECT TaxpayerID

FROM client\_table WHERE TaxpayerID = 943270000),

(SELECT StockID

FROM stock\_table WHERE StockID = 21345),

'450');

**--CREATING MUTUAL TABLE**

CREATE TABLE mutual\_table (

MutualFundID VARCHAR(255) NOT NULL,

FundName VARCHAR (255) NOT NULL,

PrincipalObjective VARCHAR (1) NOT NULL,

PriceRangeHigh Decimal(38,2) NOT NULL,

PrinceRangeLow Decimal(38,2) NOT NULL,

CurrentOffering Decimal(38,2) NOT NULL,

Yield Decimal(38,2) NOT NULL,

FamilyFundID INT,

CONSTRAINT MutualFundID\_pk PRIMARY KEY (MutualFundID),

CONSTRAINT PrincipalObjective\_Check CHECK (PrincipalObjective='G' OR PrincipalObjective='I' OR PrincipalObjective='G') );

**--INSERTING VALUES INTO MUTUAL TABLE**

INSERT INTO mutual\_table (

MutualFundID,

FundName,

PrincipalObjective,

PriceRangeHigh,

PrinceRangeLow,

CurrentOffering,

Yield,

FamilyFundID )

VALUES

('2345',

'ABC Fund',

'G',

5500.67,

1100.89,

4700.11,

12,

110

);

INSERT INTO mutual\_table (

MutualFundID,

FundName,

PrincipalObjective,

PriceRangeHigh,

PrinceRangeLow,

CurrentOffering,

Yield,

FamilyFundID )

VALUES

('7675',

'DEF Fund',

'G',

6500.77,

1700.19,

3400.19,

14,

111

);

INSERT INTO mutual\_table (

MutualFundID,

FundName,

PrincipalObjective,

PriceRangeHigh,

PrinceRangeLow,

CurrentOffering,

Yield,

FamilyFundID )

VALUES

('8765',

'GHI Fund',

'I',

2200,

1900,

2050,

58,

112

);

INSERT INTO mutual\_table (

MutualFundID,

FundName,

PrincipalObjective,

PriceRangeHigh,

PrinceRangeLow,

CurrentOffering,

Yield,

FamilyFundID )

VALUES

('5456',

'JKL Fund',

'I',

1200,

1050,

1050,

87,

113

);

INSERT INTO mutual\_table (

MutualFundID,

FundName,

PrincipalObjective,

PriceRangeHigh,

PrinceRangeLow,

CurrentOffering,

Yield,

FamilyFundID )

VALUES

('4956',

'XYZ Fund',

'G',

8200,

7600,

7900,

80,

114

);

**--CREATING FAMILY TABLE**

CREATE TABLE family\_table (

FamilyFundID INT NOT NULL,

MutualFundID VARCHAR (255) NOT NULL,

FullName VARCHAR (255) NOT NULL,

Street VARCHAR (255) NOT NULL,

City VARCHAR (255) NOT NULL,

State VARCHAR (255) NOT NULL,

Zip INT NOT NULL,

Country VARCHAR (255) NOT NULL,

CONSTRAINT FamilyFundID\_pk PRIMARY KEY (FamilyFundID) ,

CONSTRAINT MutualFundID\_fk FOREIGN KEY (MutualFundID) REFERENCES mutual\_table (MutualFundID) on delete cascade);

**--INSERTING VALUES IN FAMILY TABLE**

INSERT INTO family\_table (

FamilyFundID,

MutualFundID,

FullName,

Street,

City,

State,

Zip,

Country

)

VALUES

((SELECT FamilyFundID

FROM mutual\_table WHERE FamilyFundID = 110),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 2345),

'Jose Ulloa',

'435 Fake Street',

'Los Angeles',

'CA',

'12345',

'USA');

INSERT INTO family\_table (

FamilyFundID,

MutualFundID,

FullName,

Street,

City,

State,

Zip,

Country

)

VALUES

((SELECT FamilyFundID

FROM mutual\_table WHERE FamilyFundID = 111),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 7675),

'Mike Smith',

'1878 Noreal Blvd',

'Los Angeles',

'CA',

'45678',

'USA');

INSERT INTO family\_table (

FamilyFundID,

MutualFundID,

FullName,

Street,

City,

State,

Zip,

Country

)

VALUES

((SELECT FamilyFundID

FROM mutual\_table WHERE FamilyFundID = 112),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 8765),

'Eric Zhu',

'9898 Happy Dr',

'Miami',

'FL',

'76543',

'USA');

INSERT INTO family\_table (

FamilyFundID,

MutualFundID,

FullName,

Street,

City,

State,

Zip,

Country

)

VALUES

((SELECT FamilyFundID

FROM mutual\_table WHERE FamilyFundID = 113),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 5456),

'Aron Mejia',

'95 Spark Street',

'Seattle',

'WA',

'88765',

'USA');

INSERT INTO family\_table (

FamilyFundID,

MutualFundID,

FullName,

Street,

City,

State,

Zip,

Country

)

VALUES

((SELECT FamilyFundID

FROM mutual\_table WHERE FamilyFundID = 114),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 4956),

'Wendy Xiong',

'765 High Street',

'Austin',

'TX',

'12342',

'USA');

**--ALTERING MUTUAL TABLE**

ALTER TABLE mutual\_table

ADD CONSTRAINT FamilyFundID\_fk

FOREIGN KEY (FamilyFundID)

REFERENCES family\_table (FamilyFundID)

on delete cascade ;

**--CREATING CLIENTMUTUAL TABLE**

CREATE TABLE clientmutual\_table (

TaxpayerID INT NOT NULL,

MutualFundID VARCHAR (255) NOT NULL,

Quantity VARCHAR(255) NOT NULL,

CONSTRAINT TaxpayerID\_fk1 FOREIGN KEY (TaxpayerID) REFERENCES client\_table (TaxPayerID) on delete cascade ,

CONSTRAINT MutualFundID\_fk1 FOREIGN KEY (MutualFundID) REFERENCES mutual\_table (MutualFundId) on delete cascade );

**--INSERTING VALUES IN CLIENTMUTUAL TABLE**

INSERT INTO clientmutual\_table

(TaxpayerID,

MutualFundID,

Quantity)

VALUES

((SELECT TaxPayerID

FROM client\_table WHERE TaxPayerID = 545786625),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 2345),

100);

INSERT INTO clientmutual\_table

(TaxpayerID,

MutualFundID,

Quantity)

VALUES

((SELECT TaxPayerID

FROM client\_table WHERE TaxPayerID = 123456789),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 7675),

150);

INSERT INTO clientmutual\_table

(TaxpayerID,

MutualFundID,

Quantity)

VALUES

((SELECT TaxPayerID

FROM client\_table WHERE TaxPayerID = 987654321),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 8765),

200);

INSERT INTO clientmutual\_table

(TaxpayerID,

MutualFundID,

Quantity)

VALUES

((SELECT TaxPayerID

FROM client\_table WHERE TaxPayerID = 128765930),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 5456),

250);

INSERT INTO clientmutual\_table

(TaxpayerID,

MutualFundID,

Quantity)

VALUES

((SELECT TaxPayerID

FROM client\_table WHERE TaxPayerID = 943270000),

(SELECT MutualFundID

FROM mutual\_table WHERE MutualFundID = 4956),

300);

**/\* Describe tables \*/**

desc client\_table;

Name Null? Type

---------- -------- -------------

TAXPAYERID NOT NULL NUMBER(38)

FIRSTNAME NOT NULL VARCHAR2(255)

LASTNAME NOT NULL VARCHAR2(255)

STREET NOT NULL VARCHAR2(255)

CITY NOT NULL VARCHAR2(255)

STATE NOT NULL VARCHAR2(255)

ZIP NOT NULL NUMBER(38)

COUNTRY NOT NULL VARCHAR2(255)

desc stock\_table;

Name Null? Type

-------------------- -------- -------------

STOCKID NOT NULL VARCHAR2(255)

ISSUENAME NOT NULL VARCHAR2(255)

RATING NOT NULL VARCHAR2(100)

PRINCIPALBUINESS NOT NULL VARCHAR2(255)

PRICERANGEHIGH NOT NULL NUMBER(38,2)

PRINCERANGELOW NOT NULL NUMBER(38,2)

CURRENTPRICE NOT NULL NUMBER(38,2)

PRIORYEARRETURN NUMBER(38,2)

FIVEYEARPERIODRETURN NUMBER(38,2)

desc clientstock\_table;

Name Null? Type

---------- -------- -------------

TAXPAYERID NOT NULL NUMBER(38)

STOCKID NOT NULL VARCHAR2(255)

QUANTITY NOT NULL VARCHAR2(255)

desc mutual\_table;

Name Null? Type

------------------ -------- -------------

MUTUALFUNDID NOT NULL VARCHAR2(255)

FUNDNAME NOT NULL VARCHAR2(255)

PRINCIPALOBJECTIVE NOT NULL VARCHAR2(1)

PRICERANGEHIGH NOT NULL NUMBER(38,2)

PRINCERANGELOW NOT NULL NUMBER(38,2)

CURRENTOFFERING NOT NULL NUMBER(38,2)

YIELD NOT NULL NUMBER(38,2)

FAMILYFUNDID NUMBER(38)

desc family\_table;

Name Null? Type

------------ -------- -------------

FAMILYFUNDID NOT NULL NUMBER(38)

MUTUALFUNDID NOT NULL VARCHAR2(255)

FULLNAME NOT NULL VARCHAR2(255)

STREET NOT NULL VARCHAR2(255)

CITY NOT NULL VARCHAR2(255)

STATE NOT NULL VARCHAR2(255)

ZIP NOT NULL NUMBER(38)

COUNTRY NOT NULL VARCHAR2(255)

desc clientmutual\_table;

Name Null? Type

------------ -------- -------------

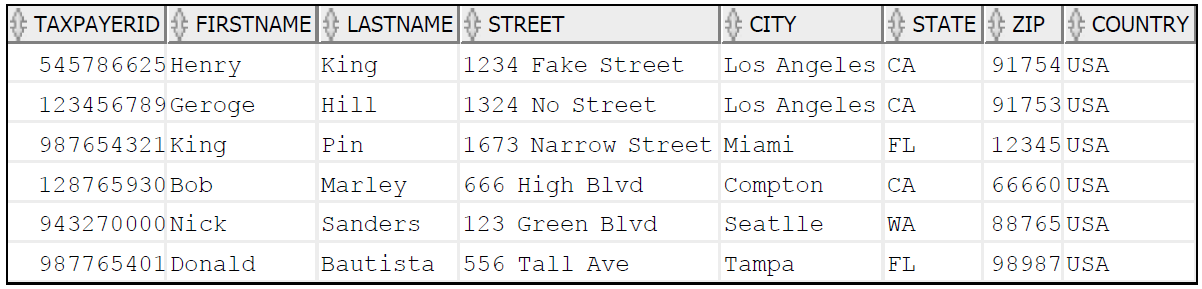
TAXPAYERID NOT NULL NUMBER(38)

MUTUALFUNDID NOT NULL VARCHAR2(255)

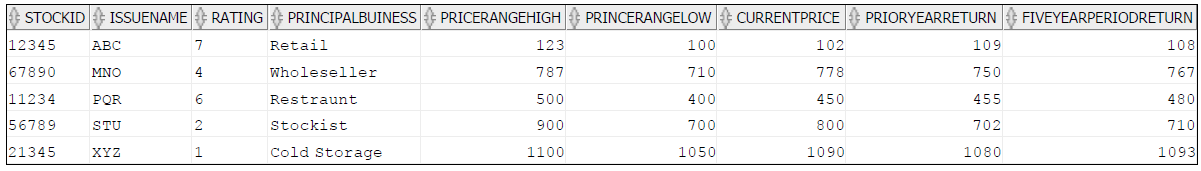
QUANTITY NOT NULL VARCHAR2(255)

**/\* Select Table Statements \*/**

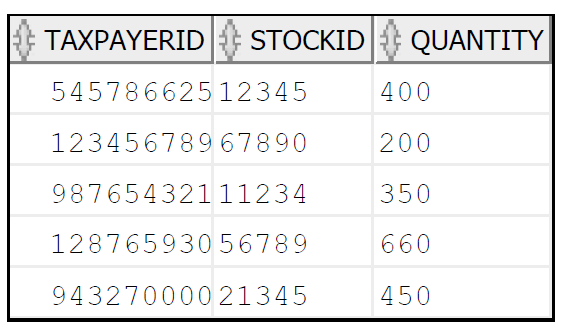
SELECT \* FROM client\_table;



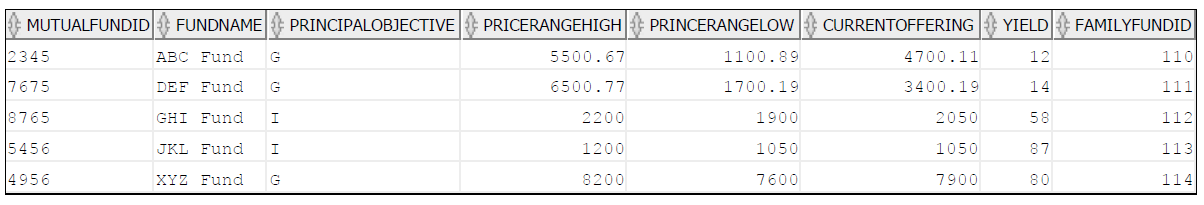
SELECT \* FROM stock\_table;



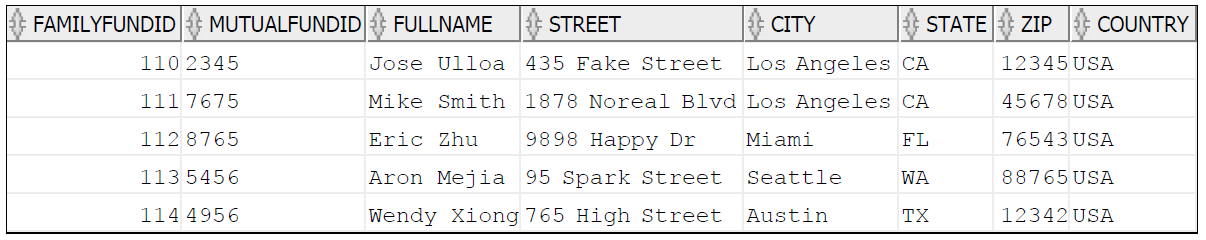
SELECT \* FROM clientstock\_table;



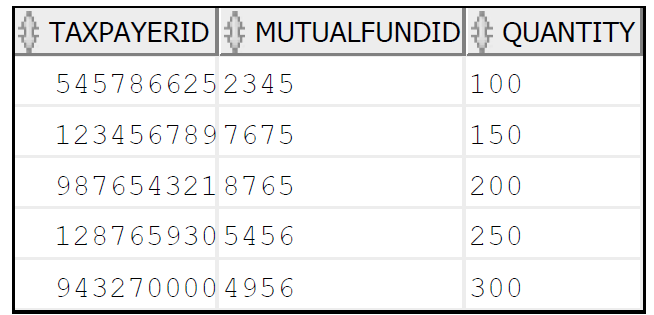
SELECT \* FROM mutual\_table;



SELECT \* FROM family\_table;

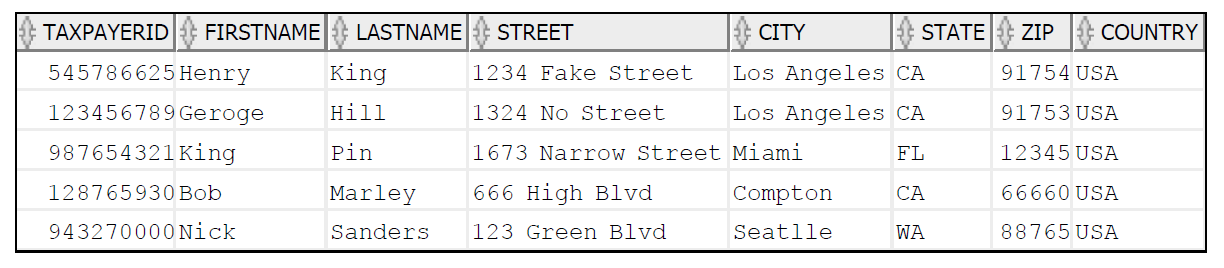


SELECT \* FROM clientmutual\_table;

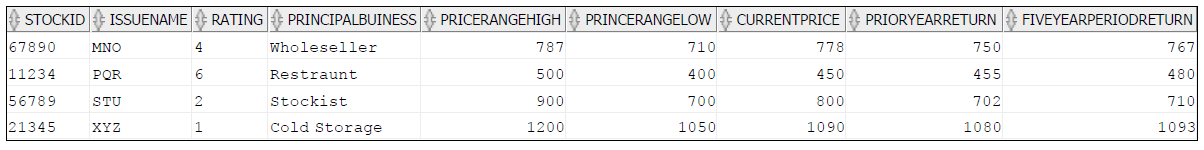


**/\* Delete statements \*/**

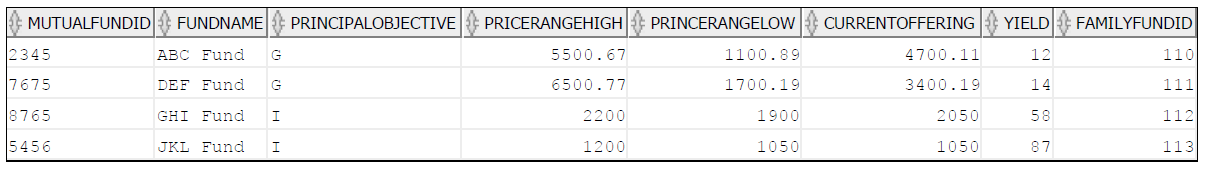
delete from client\_table where Taxpayerid=987765401;



delete from stock\_table where Rating=7;



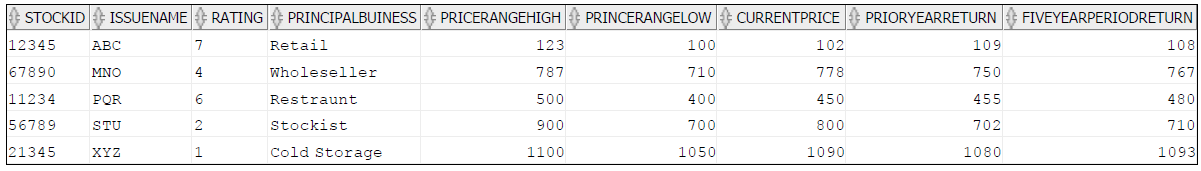
delete from mutual\_table where Mutualfundid=4956;



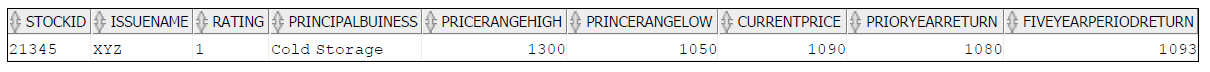
**/\* Update statements \*/**

update stock\_table set pricerangehigh=1300 where stockid=21345;

Before:

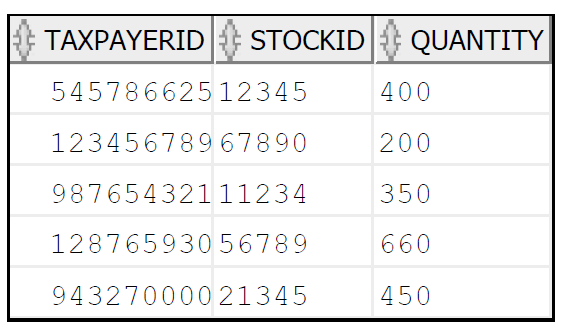


After: select \* from stock\_table where stockid=21345;

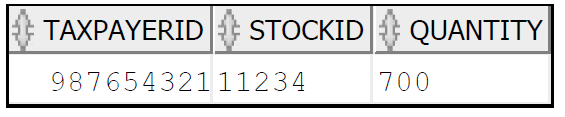


update clientstock\_table set quantity=700 where stockid=11234;

Before:



After: select \* from clientstock\_table where stockid=11234;



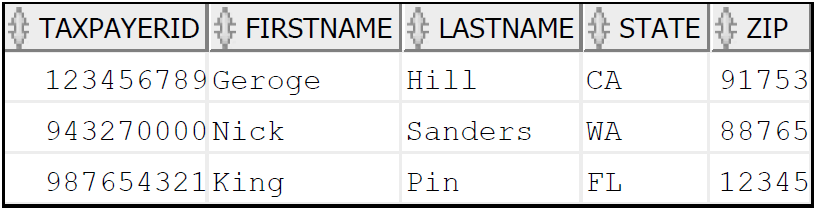
**/\* Views \*/**

create or replace view client\_view

AS (select taxpayerid,firstname,lastname,state,zip from client\_table where taxpayerid in(943270000,987654321,123456789 ));

View CLIENT\_VIEW created.

select \* from client\_view;

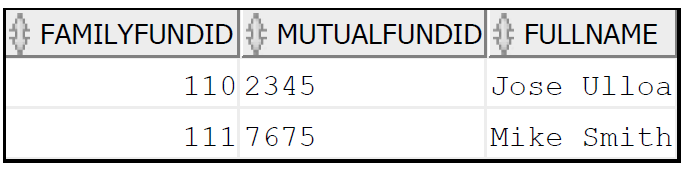


create or replace view family\_view

As ( select familyfundid, mutualfundid, fullname from Family\_Table where city='Los Angeles');

View FAMILY\_VIEW created.

select \* from family\_view;

­­

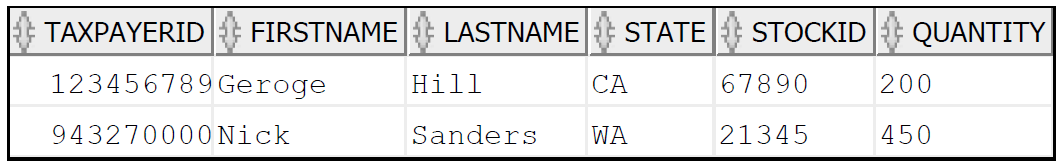
**/\* Complex Select Statements, JOIN, GROUP BY, ETC DML \*/**

select c.taxpayerid , c.firstname, c.lastname, c.state, s.stockid, s.quantity

from client\_table c, clientstock\_table s

where c.taxpayerid=s.taxpayerid

having s.quantity between 200 and 450 group by c.taxpayerid,c.firstname,c.lastname,c.state, s.stockid, s.quantity;



select \* from

(select b.stockid,a.firstname,a.lastname, b.principalbuiness, b.rating, b.currentprice

from clientstock\_table c,stock\_table b,client\_table a

where b.stockid =c.stockid and a.taxpayerid=c.taxpayerid)

where principalbuiness='Wholeseller' --- Inline views

order by currentprice

