

\*\*SUBMITTED TO NORTHEASTERN UNIVERSITY

# WEALTH MANAGEMENT

FOR HIGH NETWORTH INDIVIDUALS

SERVICES INCLUDE:

- 1) WEALTH AND LEGACY PLANNING
- 2) LENDING STRATEGIES AND LIQUIDITY MANAGEMENT
- 3) FAMILY ENTERPRISE
- 4) PHILANTHROPY
- 5) ETC..

# PERSONAL BANKING

FOR COMMON INDIVIDUALS

SERVICES INCLUDE:

- 1) CREATION OF CHECKING/SAVINGS ACCOUNT
- 2) DEBIT/CREDIT CARDS
- 3) LOANS
- 4) ETC..



# ASSET MANAGEMENT

FOR CORPORATES AND COMMON INDIVIDUALS

SERVICES INCLUDE:

- 1) MULTI ASSET INVESTING
- 2) SUSTAINABILITY INVESTING
- 3) MUNICIPAL BONDS
- 4) ETC..

# INVESTMENT BANKING

FOR CORPORATES

SERVICES INCLUDE:

- 1) FINANCIAL ADVISING
- 2) MANAGE INITIAL PUBLIC OFFERING BY UNDERWRITING

# DATABASE MANAGEMENT FOR ANALYTICS

WHY DO WE NEED TO PERFORM ANALYTICS ON DATABASE CREATED FOR A FINANCIAL SERVICES INSTITUTION?

TO ANSWER QUESTIONS LIKE -

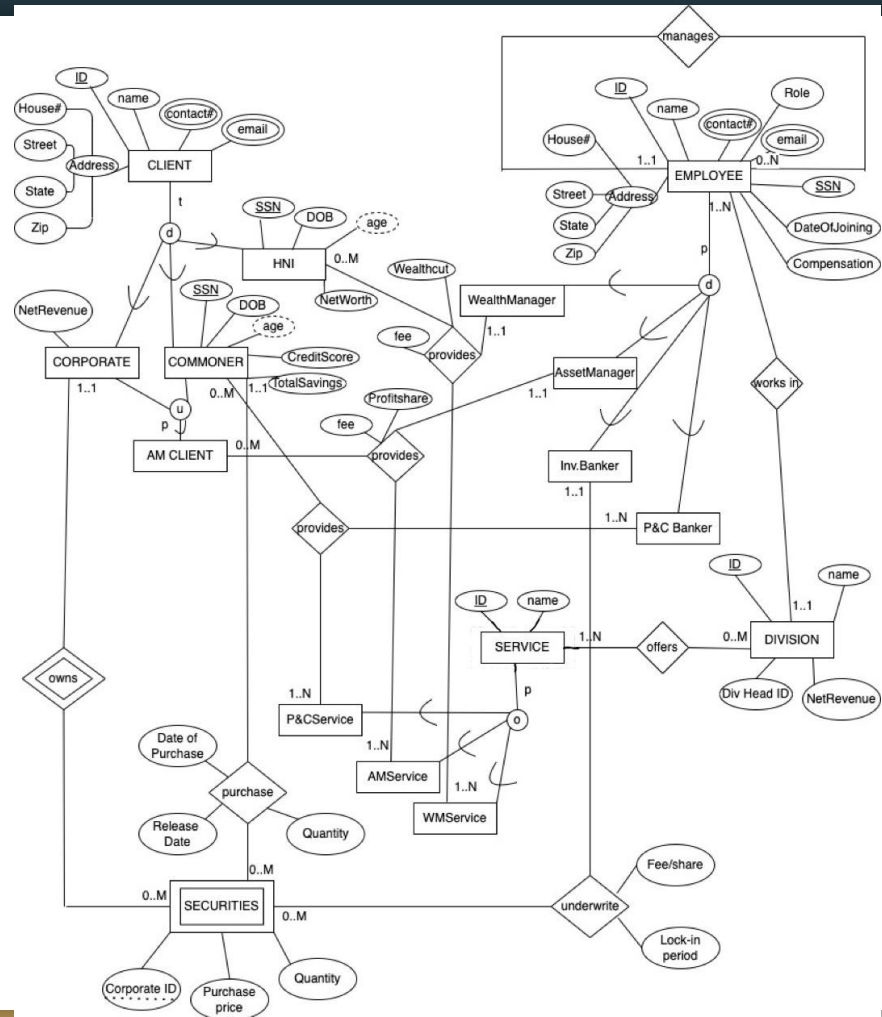
- 1) WHO ALL ARE THE CLIENTS? ARE THE DIFFERENT CLASSES OF CLIENTS GETTING THE BEST SERVICES ACCORDING TO THEIR NEEDS?
- 2) HOW IS EACH DIVISION OF THE FIRM DOING?
- 3) HOW MUCH REVENUE IS EACH DIVISION BRINGING TO THE FIRM?
- 4) ARE EMPLOYEES GETTING FAIRLY COMPENSATED?
- 5) WHO ARE THE FIRM'S LOYAL CLIENTS?
- 6) CAN THE REVENUE OF EACH DIVISION IN THE FUTURE BE PREDICTED?

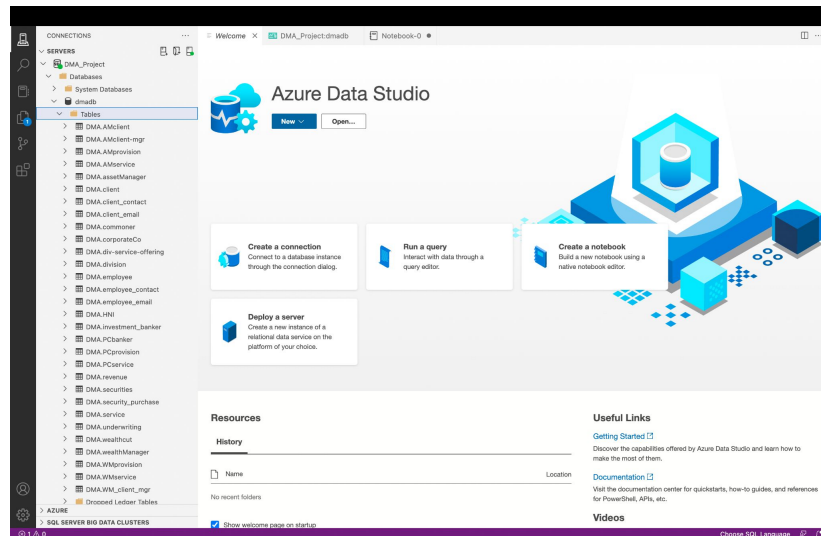
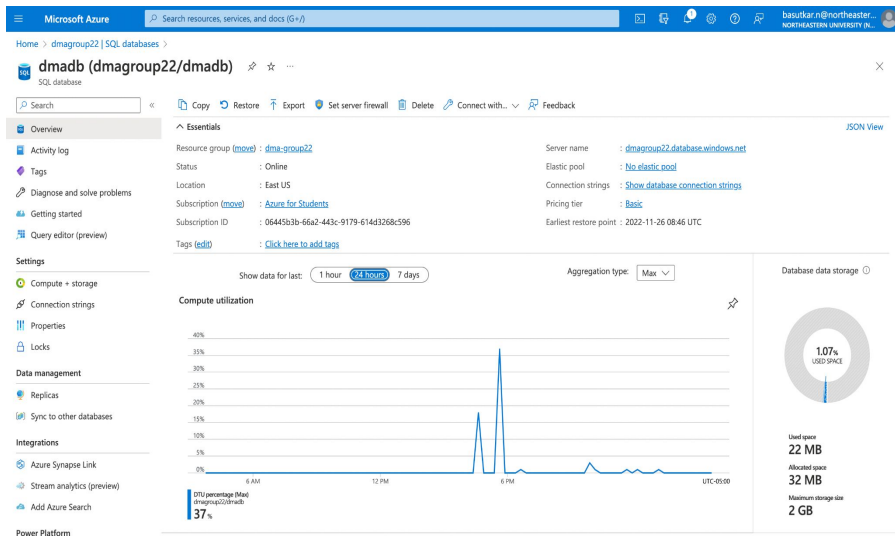


# ENHANCED ENTITY-RELATIONSHIP MODEL

## POINTS TO NOTE:

- 1) CLIENTS ARE SPECIALISED AS HNI, COMMONER AND CORPORATE WITH TOTAL AND DISJOINT RULE.
- 2) EMPLOYEES ARE SPECIALISED AS WEALTH MANAGERS/BANKERS WITH PARTIAL AND DISJOINT RULE.
- 3) CORPORATE AND COMMONER ARE CATEGORISED AS AM CLIENT.
- 4) SERVICES ARE SPECIALISED WITH PARTIAL AND OVERLAP RULE.
- 5) SECURITIES IS WEAK ENTITY TYPE DEPENDENT OF THE STRONG ENTITY TYPE - CORPORATE.





## AZURE PORTAL TO MANAGE THE DB SERVER (HANDLED BY THE DBA)

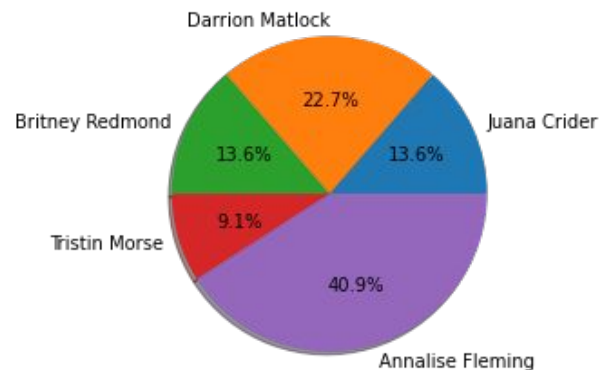
## Azure Data Studio - Database GUI tool for Data Stewards

# ANALYTICAL QUERIES - 1

WEALTH MANAGERS GET A ONE TIME WEALTH CUT FROM THE HNI CLIENTS AND CHARGE FEES FOR EACH OF THE SERVICES THEY PROVIDE. THE REVENUE WILL BE THE SUM OF THE WEALTH CUT AND FEES. CALCULATE HOW MUCH EACH HNI HAS PAID TO THE FIRM.

```
with total_fees as (  
  select HNI_id, sum(fee) as total_fees_collected  
  from dma.WMprovision  
  GROUP by HNI_id)  
select wc.HNI_id, (tf.total_fees_collected + wc.wealthcut) as  
total_hni_pay  
from dma.wealthcut wc join total_fees tf on  
wc.HNI_id=tf.HNI_id;
```

	HNI_id	name	total_hni_pay
1	1	Juana Crider	30014918
2	2	Darrion Matlock	50025008
3	3	Britney Redmond	30027761
4	4	Tristin Morse	20015599
5	5	Annalise Fleming	90023574



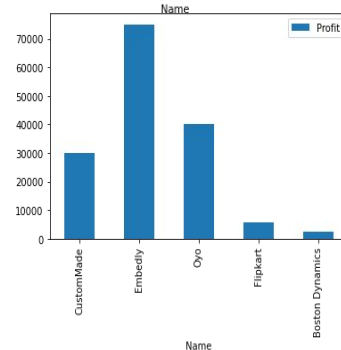
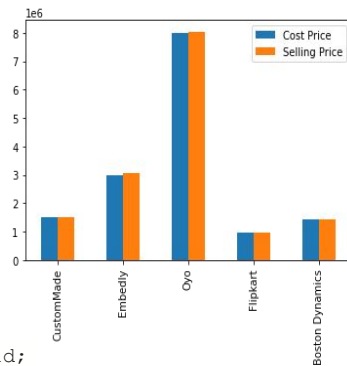
Total: \$ 220,106,860.00

# ANALYTICAL QUERIES - 2

INVESTMENT BANKERS UNDERWRITE SECURITIES OF CORPORATE COMPANIES AND SELL THEIR STOCKS. THEY CHARGE A FEE/SHARE FOR UNDERWRITING. THE REVENUE BROUGHT IN BY THIS DIVISION WILL BE THE (SELLING PRICE - COST PRICE)\* QUANTITY. ASSUMING ALL THE SECURITIES ARE SOLD, SHOW THE COMPANY NAME, ITS NET REVENUE, TOTAL SELLING PRICE, TOTAL COST PRICE AND THE PROFIT INVESTMENT BANKERS MAKE ON THE SALE.

```
with IB_stock_sell_price as ( select s.security_id,s.corporateCo_id,
(s.price_per_share + cast(u.fee_per_share as int))*s.total_quantity as sell_price
from dma.securities s join dma.underwriting u on s.security_id = u.security_id)
select sp.corporateCo_id, c.name, co.net_revenue, sp.sell_price,
(s.price_per_share * cast(s.total_quantity as int)) as cost_price,
(sp.sell_price - (s.price_per_share * cast(s.total_quantity as int))) as total_profit
from IB_stock_sell_price sp join dma.client c on sp.corporateCo_id = c.ID
join dma.securities s on sp.security_id = s.security_id join dma.corporateCo co on c.ID = co.client_id;
```

	corporateCo_id	name	net_revenue	sell_price	cost_price	total_profit
1	26	CustomMade	82000000	1530000	1500000	30000
2	27	Embedly	62000000	3075000	3000000	75000
3	28	Oyo	42000000	8040000	8000000	40000
4	29	Flipkart	22000000	985600	980000	5600
5	30	Boston Dynamics	20000000	1442400	1440000	2400





# CONNECTING TO DATABASE USING CONNECTION STRING IN PYTHON

```
import pyodbc
server = 'dmagroup22.database.windows.net'
database = 'dmadb'
username = 'CloudSA169ab7fd'
password = '{DMAGroup22}'
driver= '{ODBC Driver 17 for SQL Server}'
conn_string = 'DRIVER='+driver+';SERVER=tcp:'+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+password
```

3) Find the amount of money brought in by the P&C division.

```
with pyodbc.connect('DRIVER='+driver+';SERVER=tcp:'+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+password) as conn:
    with conn.cursor() as cursor:
        cursor.execute("select sum(total_savings) as total_inflow from dma.commoner;")
        columns = [column[0] for column in cursor.description]
        print(columns)
        row = cursor.fetchone()
        while row:
            print (str(row[0]))
            row = cursor.fetchone()

['total_inflow']
951009
```



# NOSQL- MONGODB

## INSTALLATION COMMANDS ON MAC-

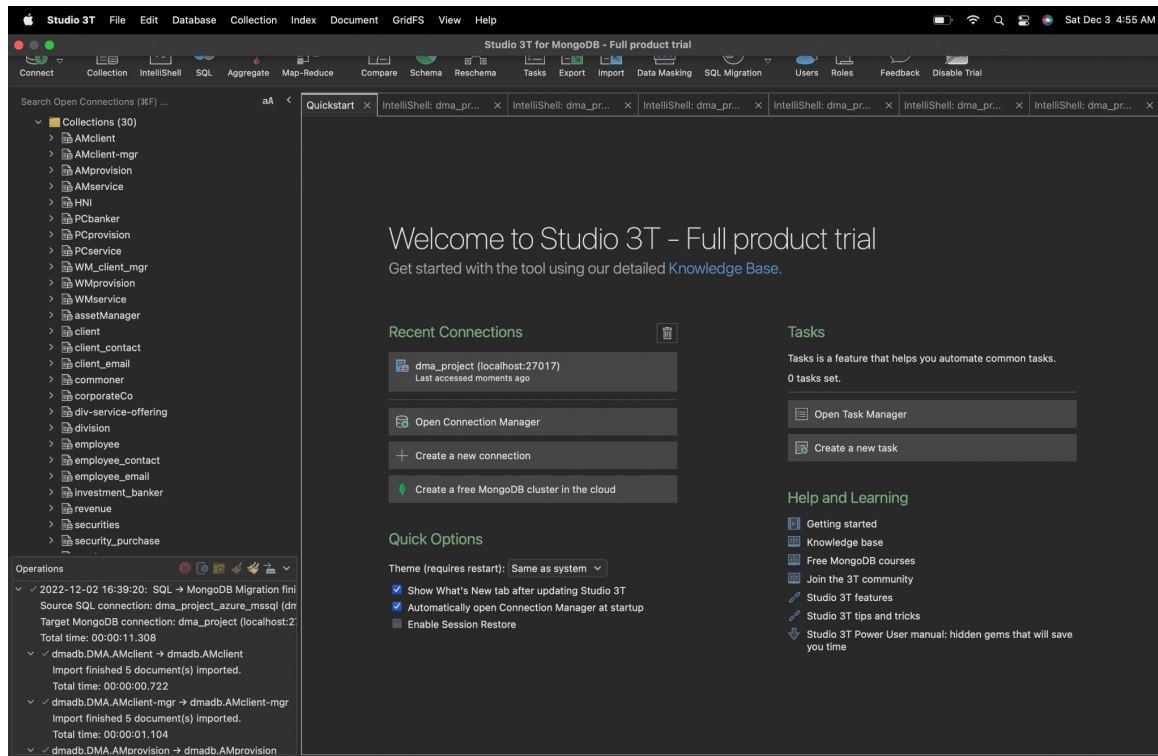
```
brew tap mongodb/brew
brew install
mongodb-community@6.0
```

## TO RUN MONGODB AS A SERVICE-

```
brew services start
mongodb-community@6.0
```

## TO STOP MONGODB-

```
brew services stop
mongodb-community@6.0
```



# ANALYTICAL QUERIES IN MONGODB

HOW MUCH WEALTH HAS EACH DIVISION  
ACCUMULATED OVER THE YEARS?

```
1 db.revenue.aggregate([
2   $group: {
3     _id: "$div_id",
4     total_revenue: {
5       $sum: "$revenue"
6     }
7   }
8 ]]);
```

Raw Shell Output | Shell Output (Array) ×

← → 50 Documents 1 to 1

Array > element objects ("\$\$")

[Index]	total_revenue
0	81000000000
1	99000000000
2	13500000000
3	162000000000

GET THE DETAILS OF THE INVESTMENT BANKERS WHO HAVE  
UNDERWRITTEN TILL NOW.

```
1 var underwriters = db.underwriting.distinct("IBanker_id")
2
3 db.employee.find({"ID": {"$in": underwriters}});
4
```

Raw Shell Output | Find Query (line 3) ×

← → 50 Documents 1 to 2

employee

ID	SSN	name	role
1	889599561	Vishal Basutkar	Investment Bank
6	205602683	Keenan Lieberman	Investment Bank

HOW MANY EMPLOYEES WORK FOR EACH  
MANAGER?

```
1 db.employee.aggregate([
2   $match: {manager_id: {$exists: true, $ne: null}},
3   $group: { _id: "$manager_id",
4     num_of_employees: {$count: {}} }
5 ]]);
```

Raw Shell Output | Aggregate Query (line 1) ×

← → 50 Documents 1 to 4

employee > num\_of\_employees

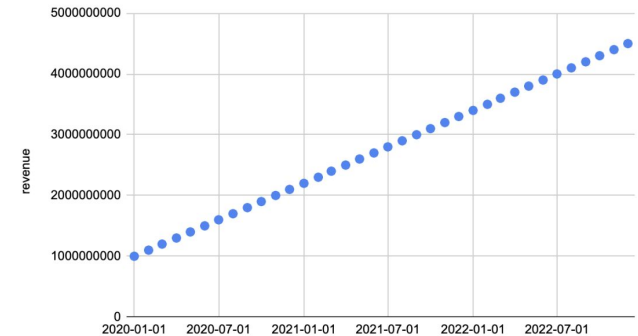
_id	num_of_employees
7	6
5	7
10	6
4	7

# PERFORMING MACHINE LEARNING ON DATA RETRIEVED FROM THE REVENUE TABLE IN THE DATABASE TO PREDICT THE REVENUE OF WEALTH MANAGEMENT DIVISION IN JAN 2023.

```
[27] #retrieving Wealth Management Revenue Data from database and saving in a dataframe.
wm_df = pd.DataFrame()
with pyodbc.connect(conn_string) as conn:
    with conn.cursor() as cursor:
        print('connection created')
        cursor.execute("select * from dma.revenue where div_id=1")
        row = cursor.fetchone()
        i=0
        while row:
            wm_df.loc[i,'date'] = str(row[1])
            wm_df.loc[i,'revenue'] = row[2]
            i+=1
            row = cursor.fetchone()
print(wm_df)
wm_df_temp = wm_df
wm_df_temp.plot(x="date", y="revenue", kind='scatter')
```

```
#converting string type date to datetime and then to ordinal.
import datetime as dt
wm_df_temp['date'] = pd.to_datetime(wm_df_temp['date'])
wm_df_temp['date']=wm_df_temp['date'].map(dt.datetime.toordinal)
wm_df_temp
```

WM revenue



```
[29] X = np.array(wmdf_temp['date']).reshape(-1, 1)
      y = np.array(wmdf_temp['revenue']).reshape(-1, 1)

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
      # Splitting the data into training and testing data

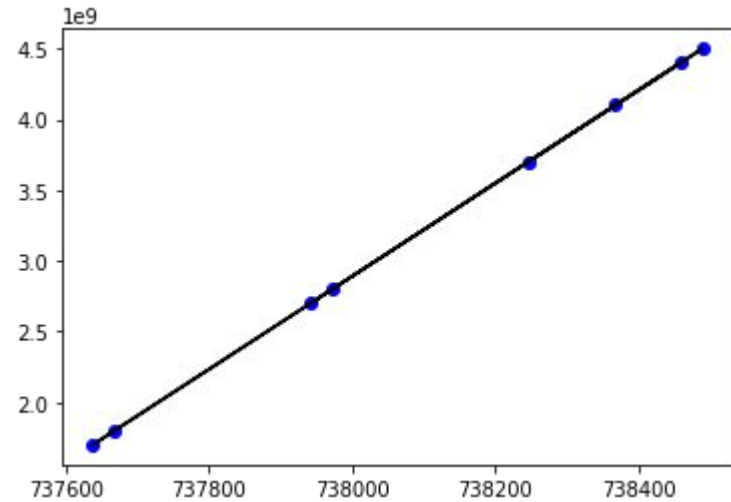
      regr = LinearRegression()

      regr.fit(X_train, y_train)
      print("Model Accuracy: ", regr.score(X_test, y_test))
```

Model Accuracy: 0.999998240528332

```
▶ y_pred = regr.predict(X_test)
  plt.scatter(X_test, y_test, color='b')
  plt.plot(X_test, y_pred, color='k')

  plt.show()
```



```
▶ predicted_revenue = regr.predict([[pd.to_datetime('2023-01-01').toordinal()]])
  print(predicted_revenue[0,0])
```

➞ 4602791180.799805



THANK YOU!