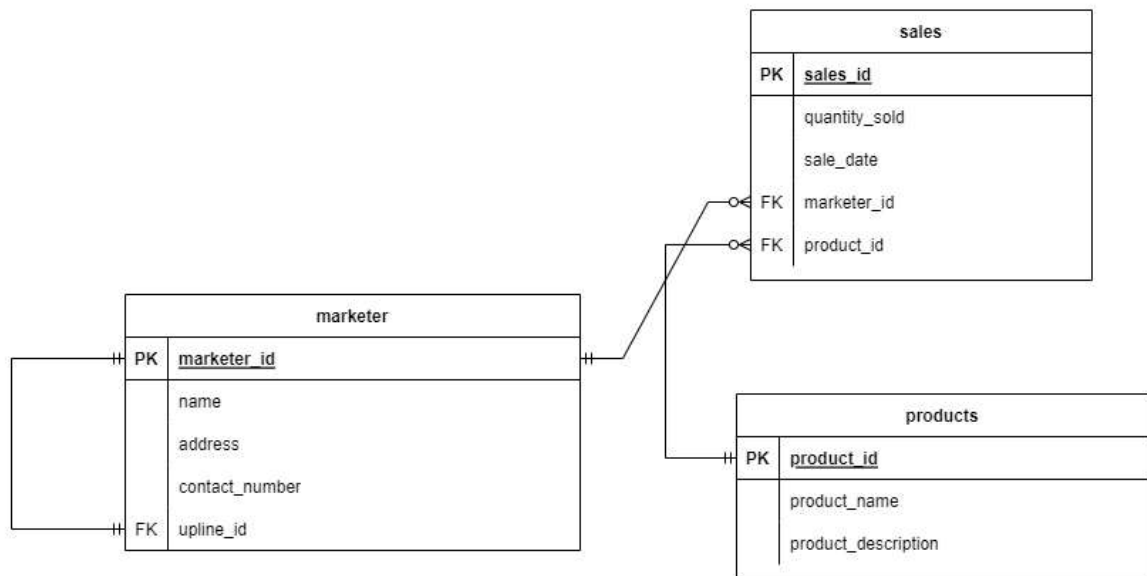


DEDS: Exercise 2

Vincent Itucal

Updated ERD

Removed the network tables and created self referencing foreign key `upline_id` that points to the primary key `marketer_id` representing the upline



```
In [40]: from faker import Faker
import random
import pandas as pd
from datetime import datetime, timedelta
from sqlalchemy import create_engine
```

```
In [1]: %load_ext sql
```

```
In [4]: with open('db_pass.txt', 'r') as file:
db_password = file.read().strip()
```

```
In [21]: connection_string = f"postgresql://vincent:{db_password}@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/m
```

```
In [51]: engine = create_engine(connection_string)
```

```
In [22]: get_ipython().run_line_magic('sql', connection_string)
```

Connecting to 'postgresql://vincent:***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/mlm'

Create Table Scripts

```
In [58]: %%sql
CREATE TABLE IF NOT EXISTS marketer (
    marketer_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    address TEXT,
    contact_number VARCHAR(50),
    upline_id INT,
    FOREIGN KEY (upline_id) REFERENCES marketer(marketer_id)
```

```
);

CREATE TABLE IF NOT EXISTS products (
  product_id SERIAL PRIMARY KEY,
  product_name VARCHAR(100) NOT NULL,
  product_description TEXT
);

CREATE TABLE IF NOT EXISTS sales (
  sales_id SERIAL PRIMARY KEY,
  marketer_id INT NOT NULL,
  product_id INT NOT NULL,
  quantity_sold INT NOT NULL,
  sale_date DATE NOT NULL,
  FOREIGN KEY (marketer_id) REFERENCES marketer(marketer_id),
  FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

Running query in 'postgresql://vincent:***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/mlm'

Out[58]:

List of Databases

In [23]: `%sql \l`

Running query in 'postgresql://vincent:***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/mlm'

Out[23]:

Name	Owner	Encoding	Collate	Ctype	Access privileges
mlm	vincent	UTF8	en_US.UTF-8	en_US.UTF-8	None
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	None
rdsadmin	rdsadmin	UTF8	en_US.UTF-8	en_US.UTF-8	rdsadmin=CTc/rdsadmin
template0	rdsadmin	UTF8	en_US.UTF-8	en_US.UTF-8	=c/rdsadmin rdsadmin=CTc/rdsadmin
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres postgres=CTc/postgres

List of tables in mlm database

In [46]: `%sql \dt`

Running query in 'postgresql://vincent:***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/mlm'

Out[46]:

Schema	Name	Type	Owner
public	marketer	table	vincent
public	products	table	vincent
public	sales	table	vincent

Schema of tables

In [47]: `%sqlcmd columns --table marketer`

```
Out[47]:
```

	name	type	nullable	default	autoincrement	comment
	marketer_id	INTEGER	False	nextval('marketer_marketer_id_seq'::regclass)	True	None
	name	VARCHAR(100)	False	None	False	None
	address	TEXT	True	None	False	None
	contact_number	VARCHAR(15)	True	None	False	None
	upline_id	INTEGER	True	None	False	None

```
In [48]: %sqlcmd columns --table products
```

```
Out[48]:
```

	name	type	nullable	default	autoincrement	comment
	product_id	INTEGER	False	nextval('products_product_id_seq'::regclass)	True	None
	product_name	VARCHAR(100)	False	None	False	None
	product_description	TEXT	True	None	False	None

```
In [49]: %sqlcmd columns --table sales
```

```
Out[49]:
```

	name	type	nullable	default	autoincrement	comment
	sales_id	INTEGER	False	nextval('sales_sales_id_seq'::regclass)	True	None
	marketer_id	INTEGER	False	None	False	None
	product_id	INTEGER	False	None	False	None
	quantity_sold	INTEGER	False	None	False	None
	sale_date	DATE	False	None	False	None

Faker to create dummy data

```
In [59]: fake = Faker()

# Number of records to generate
num_marketers = 100
num_products = 50
num_sales = 500

# Create marketers
marketers = []
for i in range(1, num_marketers + 1):
    marketers.append({
        'marketer_id': i,
        'name': fake.name(),
        'address': fake.address(),
        'contact_number': fake.phone_number(),
        'upline_id': random.randint(1, i - 1) if i > 1 else None # Ensure valid upline_id or None
    })

# Create products
products = []
for i in range(1, num_products + 1):
    products.append({
        'product_id': i,
        'product_name': fake.word().capitalize(),
        'product_description': fake.text(max_nb_chars=200)
    })

# Create sales
sales = []
for _ in range(num_sales):
    marketer_id = random.randint(1, num_marketers) # Random marketer_id
```

```

product_id = random.randint(1, num_products) # Random product_id
quantity_sold = random.randint(1, 100) # Random quantity sold
sale_date = fake.date_this_year() # Random sale date within this year
sales.append({
    'sales_id': len(sales) + 1,
    'marketer_id': marketer_id,
    'product_id': product_id,
    'quantity_sold': quantity_sold,
    'sale_date': sale_date
})

df_marketers = pd.DataFrame(marketers)
df_products = pd.DataFrame(products)
df_sales = pd.DataFrame(sales)

df_marketers.to_sql('marketer', engine, if_exists='append', index=False)
df_products.to_sql('products', engine, if_exists='append', index=False)
df_sales.to_sql('sales', engine, if_exists='append', index=False)

print("done faker")

```

done faker

First 100 rows of each table

In [63]: `pd.read_sql("select * from marketer limit 100", engine)`

Out[63]:

	marketer_id	name	address	contact_number	upline_id
0	1	Mr. Ryan Price	076 Carla Ports Suite 303\nRobertmouth, AL 22840	475-845-4419x61753	NaN
1	2	Kathleen Nelson	9869 Karen Pine\nPort Mollytown, FL 09344	4578728774	1.0
2	3	Jasmine Love	Unit 8359 Box 3189\nDPO AE 34026	+1-560-693-6854	1.0
3	4	Paige Wood	085 Lisa Passage\nSouth Nathanmouth, TN 04291	844.628.2552x7659	3.0
4	5	Tiffany Smith	3082 Johnson Lock\nGrahamside, VT 38390	578.662.6328	3.0
...
95	96	Taylor Williams	382 Scott Shores\nWest Stevenview, UT 58738	4553832082	37.0
96	97	James Moore	794 Johnson Loop Suite 431\nPort James, MI 97423	913-625-3390x399	38.0
97	98	Eddie Moyer	32939 Stephanie Prairie\nNorth Jennifer, RI 13133	001-443-989-9796x30588	46.0
98	99	Justin Vargas	6668 Brian Circles\nSouth Carolyn, RI 74585	229.666.6975	16.0
99	100	Brian Hernandez	7955 Ellis Roads\nNew Rossberg, DE 83165	359-646-5861x71572	25.0

100 rows × 5 columns

In [64]: `pd.read_sql("select * from products limit 100", engine)`

Out[64]:

	product_id	product_name	product_description
0	1	Lose	System campaign risk heavy. Suggest goal thous...
1	2	Back	Already possible treat throw human edge answer...
2	3	Imagine	Far major role know. Arrive general time busin...
3	4	Marriage	Trade enjoy break see. Reason wide office gard...
4	5	Cut	Stay society writer. Table prepare court bad M...
5	6	Building	Safe must another throughout dark. Its state s...
6	7	Middle	Start arm model out since scene.\nLikely list ...
7	8	Magazine	Main he lose only build organization human. AL...
8	9	Wife	Lawyer next ask seven. Join natural operation ...
9	10	Partner	Will special lot alone. Serve TV live simply f...
10	11	Soldier	Material image organization give. Friend natio...
11	12	Moment	Road energy see house. Available suddenly foll...
12	13	Music	Simple need office agree eye. Little field up ...
13	14	Water	Condition final happen. Fast decision recent s...
14	15	Fire	Play page work management director.\nNatural p...
15	16	Consider	Here level site local best able artist. Whom f...
16	17	Yet	Tonight often all adult. Cause one activity ta...
17	18	Particularly	Dark federal speak. Relate list offer reduce c...
18	19	Four	Former center affect realize fill. Rock almost...
19	20	Result	Road newspaper apply blood bank listen final. ...
20	21	Election	Will Democrat at for car model large. Kitchen ...
21	22	Answer	Bad a me land age defense official. Sense blue...
22	23	Whole	Nor price whole employee couple total. Star ci...
23	24	Research	Loss he stay executive. Difficult record write...
24	25	Safe	Character tough health environmental position ...
25	26	Particular	Current full mouth worker property sea since. ...
26	27	At	Stock somebody skill himself process concern. ...
27	28	Article	Care different be direction study. Light missi...
28	29	Significant	Story somebody tend husband. Image economy qui...
29	30	Along	Thus western management interesting. Mean if a...
30	31	Cause	Wish lot together charge. Oil thank husband.\n...
31	32	Statement	Get continue different tell. Benefit ask adult...
32	33	Public	Draw school tough safe. Chair become return co...
33	34	Fall	Team can what world easy. Later yeah ball insi...
34	35	Me	Give test inside owner religious quickly every...
35	36	Before	Spring million answer my. Marriage key off cha...
36	37	Run	Paper four call walk. Others certain resource ...
37	38	However	Result technology building.\nEasy big heart fi...

	product_id	product_name	product_description
38	39	Plant	Choice mention market easy growth. Off and fiv...
39	40	Establish	Window answer have make. Serve resource size.\...
40	41	Picture	Attention else operation town remain collectio...
41	42	Under	Rule instead professional stand. Design operat...
42	43	Discussion	Beat baby mouth hope program through. Sort wea...
43	44	Run	Before history thousand impact different. Acco...
44	45	Enjoy	Until no compare tell indeed edge ground. But ...
45	46	Even	Good interview car protect sense staff. Marria...
46	47	Leave	Throughout give participant politics. Experien...
47	48	Condition	For think detail body claim speak. Lose role t...
48	49	Compare	Threat PM enter present indeed sea decide. Exp...
49	50	Quite	Challenge institution along new imagine coach....

In [65]: `pd.read_sql("select * from sales limit 100", engine)`

Out[65]:

	sales_id	marketer_id	product_id	quantity_sold	sale_date
0	1	93	44	79	2025-01-09
1	2	44	35	68	2025-01-04
2	3	10	46	16	2025-01-12
3	4	69	21	83	2025-01-10
4	5	36	45	96	2025-01-14
...
95	96	59	36	35	2025-01-03
96	97	49	26	85	2025-01-03
97	98	43	6	17	2025-01-14
98	99	33	7	46	2025-01-01
99	100	46	46	88	2025-01-14

100 rows × 5 columns

Number of rows of each table

In [66]: `%sql select COUNT(*) from marketer`

Running query in 'postgresql://vincent:***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/mlm'

1 rows affected.

Out[66]:

count
100

In [67]: `%sql select COUNT(*) from products`

Running query in 'postgresql://vincent:***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/mlm'

1 rows affected.

Out[67]: **count**

50

In [68]: **%sql** select COUNT(*) from sales

Running query in 'postgresql://vincent:***@sampledb.c34wyku8liz.us-east-1.rds.amazonaws.com/mlm'

1 rows affected.

Out[68]: **count**

500

Rollback/Start from scratch queries

In [62]: **# %%sql**
DROP TABLE IF EXISTS marketer, products, sales CASCADE;

In []: