

Task is to populate tables with fake data. Under a MySQL database a table structure that simulate transaction table for realistic USA network retail stores (20 stores, 100 products, 20 or more promotions and transactions). In Python we will create JSON structure and load fake data and dump into the tables. Subsequently we can do any Query.

First, we create the table in MySQL where data will be written

```
mysql> show tables;
```

```
+-----+
| Tables_in_usstores |
+-----+
| products            |
| promotions          |
| stores              |
| transactions        |
+-----+
```

4 rows in set (0.00 sec)

```
mysql> describe stores;
```

```
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| store_id   | int       | NO   | PRI | NULL    |       |
| store_name | varchar(255) | NO   |     | NULL    |       |
| location   | varchar(255) | YES  |     | NULL    |       |
| manager_name | varchar(255) | YES  |     | NULL    |       |
| phone_number | varchar(255) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

5 rows in set (0.00 sec)

```
mysql> describe products;
```

```
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| product_id | int       | NO   | PRI | NULL    |       |
| product_name | varchar(255) | NO   |     | NULL    |       |
| category   | varchar(255) | YES  |     | NULL    |       |
| price      | decimal(10,2) | YES  |     | NULL    |       |
| brand      | varchar(255) | YES  |     | NULL    |       |
| description | text      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

6 rows in set (0.00 sec)

```
mysql> describe promotions;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| promotion_id | int       | NO   | PRI | NULL    | auto_increment |
| promotion_name | varchar(255) | NO   |     | NULL    |               |
| discount_percentage | decimal(5,2) | YES  |     | NULL    |               |
| start_date   | datetime  | YES  |     | NULL    |               |
| end_date     | datetime  | YES  |     | NULL    |               |
| description  | text      | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
```

6 rows in set (0.00 sec)

mysql> describe transactions;

Field	Type	Null	Key	Default	Extra
transaction_id	int	NO	PRI	NULL	
store_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
transaction_date	datetime	YES		NULL	
quantity	int	YES		NULL	
total_amount	decimal(10,2)	YES		NULL	
promotion_id	int	YES	MUL	NULL	
payment_method	varchar(50)	YES		NULL	
customer_id	int	YES		NULL	

9 rows in set (0.00 sec)

mysql>

## Python Code

```
from faker import Faker
import random
import json
import mysql.connector
from datetime import datetime, timedelta

def insert_data(table_name, data):
    try:
        #create a connection
        spcdatabase=mysql.connector.connect(
            user='root',
            host='localhost',
            password='HAHAHAHAHA',
            database='usstores'
        )
        spccur=spcdatabase.cursor()

        # Insert data into the table
        for record in data:
            placeholders = ', '.join(['%s'] * len(record))
            columns = ', '.join(record.keys())
            sql = f"INSERT INTO {table_name} ({columns}) VALUES ({placeholders})"
            spccur.execute(sql, tuple(record.values()))

        spcdatabase.commit()
        print(f"Data inserted into {table_name} successfully!")

    except Exception as e:
        print(f"Error inserting data into {table_name}: {e}")

    finally:
        spcdatabase.close()

fake = Faker()
```

```
# Define some known promotional campaign names
```

```
promotion_names = [  
    "Summer Sale",  
    "Black Friday",  
    "Holiday Special",  
    "Clearance Event",  
    "Back-to-School Discount",  
    "Christmas Sale",  
    "New Year's Sale",  
    "Spring Sale",  
    "Year End Sale",  
    "All Clearance Sale"  
]
```

```
# Generate US Stores Data
```

```
def generate_stores(num_stores):  
    stores = []  
    for store_id in range(1, num_stores + 1):  
        store = {  
            "store_id": store_id,  
            "store_name": fake.company(),  
            "location": fake.city() + ", " + fake.state_abbrev(),  
            "manager_name": fake.name(),  
            "phone_number": fake.phone_number()  
        }  
        stores.append(store)  
    return stores
```

```
# Generate Product Data
```

```
def generate_products(num_products):  
    product_categories = ["Electronics", "Clothing", "Groceries", "Home Goods", "Beauty", "Toys"]  
    products = []  
    for product_id in range(1, num_products + 1):  
        product = {  
            "product_id": product_id,  
            "product_name": fake.word(),  
            "category": random.choice(product_categories),  
            "price": round(random.uniform(10, 200), 2),  
            "brand": fake.company(),  
            "description": fake.catch_phrase()  
        }  
        products.append(product)  
    return products
```

```
def generate_promotions(num_records):
```

```
    promotions = []  
    for promotion_id in range(1, num_records + 1):  
        promotion = {  
            "promotion_id": promotion_id,  
            "promotion_name": random.choice(promotion_names),  
            "discount_percentage": round(random.uniform(5, 50), 2),  
            "start_date": (datetime.now() + timedelta(days=random.randint(1, 30))).strftime("%Y-%m-%d %H:%M:%S"),  
            "end_date": (datetime.now() + timedelta(days=random.randint(31, 60))).strftime("%Y-%m-%d %H:%M:%S"),  
            "description": fake.sentence()  
        }  
        promotions.append(promotion)
```

```

return promotions

if __name__ == "__main__":
    num_records = 20 # Adjust as needed
    num_stores = 20
    num_products = 100
    promotions_data = generate_promotions(num_records)
    stores_data = generate_stores(num_stores)
    products_data = generate_products(num_products)

    # Save data to a JSON file
    with open("promotions_data.json", "w") as json_file:
        json.dump(promotions_data, json_file, indent=2)

    with open("stores_data.json", "w") as stores_file:
        json.dump(stores_data, stores_file, indent=2)

    with open("products_data.json", "w") as products_file:
        json.dump(products_data, products_file, indent=2)

    with open("stores_data.json", "r") as stores_file:
        stores_data = json.load(stores_file)

    with open("products_data.json", "r") as products_file:
        products_data = json.load(products_file)

    with open("promotions_data.json", "r") as promotions_file:
        promotions_data = json.load(promotions_file)

    # Insert data into MySQL tables
    insert_data("stores", stores_data)
    insert_data("products", products_data)
    insert_data("promotions", promotions_data)

```

**That's it, we run it now**

**Output is**

```

Data inserted into stores successfully!
Data inserted into products successfully!
Data inserted into promotions successfully!

```

**Let's check the tables now**

## Table stores

```
mysql> select * from stores;
```

store_id	store_name	location	manager_name	phone_number
1	Mathis, Morgan and Blevins	Figueroborough, MD	Scott McKay	001-728-534-6451x253
2	Powell-Schneider	Colinside, IA	Amanda Williams	(935)827-7843
3	Elliott, Hanson and Wells	Scottnmouth, GA	Cheryl Johnson	001-836-859-5220x316
4	Burgess-Davis	Hansonchester, NM	Sarah Williams	912.379.5670x17428
5	Christensen-Griffin	Gordonland, NY	Emily Conley	331.435.1732x5050
6	Mora and Sons	New Devin, WA	Ronnie Martin	982.223.4616x4217
7	Stanton-Woodard	North Russellside, IL	Anthony Gonzales	617-971-7522
8	Gomez-Jackson	North Paulshire, RI	Martha Mills	642-781-8421x46429
9	Baker-Thompson	Johnsontown, LA	Kelly Carrillo	542.745.8509x68217
10	Clark, Li and Taylor	Pageview, VT	Jill Le	001-369-340-2428x77197
11	Foley-Wood	South Joshua, AK	Daniel Rodriguez	780-353-1240
12	Hanna, Floyd and Zamora	Taylorbury, GA	Elizabeth Jones	207-293-6125
13	Little PLC	North Antonioville, DC	Taylor Green	001-412-371-7076x405
14	Hart, Jensen and Serrano	Fisherfort, ME	Thomas Serrano	6936602983
15	Allen, Cannon and Watson	Lake Kristy, CA	Sarah Brown	+1-505-883-8385x396
16	Jones, Franklin and Travis	Martinezmouth, VA	Rachel Compton	626-398-1109
17	Moore-Everett	South Matthew, SC	Debbie Perkins	+1-445-210-2322
18	Orozco-Baker	Hoganchester, PW	Nicole Castillo	+1-456-647-4623
19	White, Reyes and James	Hawkinsmouth, WY	Stephen Rodriguez	+1-829-909-7406x45484
20	Aguirre-Hartman	Hallfort, AK	Dr. Michael Duran	590.429.6172x064

20 rows in set (0.00 sec)

## Table promotions

```
mysql> select promotion_id as id, promotion_name as name, discount_percentage as discount, start_date, end_date, description from promotions;
```

id	name	discount	start_date	end_date	description
1	Spring Sale	8.41	2024-06-08 18:52:09	2024-07-14 18:52:09	Soon discuss actually deep black.
2	New Year's Sale	6.25	2024-06-19 18:52:09	2024-06-30 18:52:09	Test story dinner dream no amount.
3	Clearance Event	31.98	2024-06-02 18:52:09	2024-07-11 18:52:09	Fill model return hotel.
4	Clearance Event	9.80	2024-05-28 18:52:09	2024-06-27 18:52:09	These study Mr now me.
5	All Clearance Sale	37.17	2024-06-22 18:52:09	2024-07-14 18:52:09	Hand no soldier lawyer dinner time.
6	Christmas Sale	12.03	2024-06-22 18:52:09	2024-06-28 18:52:09	Success remember allow understand law turn arm.
7	Clearance Event	30.57	2024-05-29 18:52:09	2024-07-05 18:52:09	Or dark discussion card most deep.
8	New Year's Sale	31.25	2024-06-09 18:52:09	2024-07-04 18:52:09	Agent defense religious television lot truth scene.
9	Year End Sale	48.54	2024-06-10 18:52:09	2024-06-28 18:52:09	Success alone type they special military should while.
10	Clearance Event	15.36	2024-06-06 18:52:09	2024-07-18 18:52:09	Hot information interest stuff reality deal form.
11	Clearance Event	30.49	2024-06-10 18:52:09	2024-07-08 18:52:09	Strong air center with sign western half conference.
12	Spring Sale	8.03	2024-06-03 18:52:09	2024-07-05 18:52:09	Finally likely anyone national him.
13	All Clearance Sale	5.41	2024-05-26 18:52:09	2024-07-08 18:52:09	Western none better than.
14	Back-to-School Discount	29.77	2024-06-12 18:52:09	2024-06-30 18:52:09	Away outside blood spend.
15	All Clearance Sale	36.61	2024-06-15 18:52:09	2024-07-15 18:52:09	Sit tell other listen box moment natural.
16	Spring Sale	48.17	2024-06-14 18:52:09	2024-07-20 18:52:09	Pretty behavior impact fine pretty detail.
17	Clearance Event	11.73	2024-06-04 18:52:09	2024-06-29 18:52:09	Sure job camera who deep.
18	Summer Sale	12.96	2024-06-10 18:52:09	2024-07-18 18:52:09	Ground edge spring anyone least people.
19	All Clearance Sale	39.96	2024-05-27 18:52:09	2024-07-20 18:52:09	Give either staff wall city yeah second toward.
20	Spring Sale	26.38	2024-06-17 18:52:09	2024-07-03 18:52:09	Us would product art own.

20 rows in set (0.00 sec)

## Table products

mysql> select \* from products;

product_id	product_name	category	price	brand	description
1	claim	Toys	124.27	Goodwin-Copeland	Horizontal radical installation
2	value	Toys	125.32	Harris, Marquez and Arias	Streamlined client-driven Graphic Interface
3	partner	Toys	13.05	Harrison Group	Synergized systemic strategy
4	ask	Beauty	198.56	Williams and Sons	Visionary methodical interface
5	shoulder	Home Goods	17.56	Salazar, Wilson and Ramirez	Face-to-face heuristic pricing structure
6	either	Toys	107.10	Fitzgerald-Stout	Synchronized executive moderator
7	make	Electronics	14.79	Stone Ltd	Triple-buffered local portal
8	attack	Toys	71.45	Pittman, Johnson and Noble	Cross-platform coherent matrices
9	idea	Groceries	61.05	Herrera, Long and Martin	Profit-focused mobile capacity
10	safe	Toys	183.59	Morrow and Sons	User-friendly clear-thinking implementation
11	mission	Electronics	174.83	Ross, Miller and Pierce	Universal uniform parallelism
12	brother	Home Goods	98.69	Anderson, Martinez and Monroe	Centralized content-based algorithm
13	seem	Toys	130.28	Moran-Logan	Reduced bandwidth-monitored capability
14	improve	Toys	115.51	Richardson-Davidson	Expanded intangible matrix
15	dinner	Home Goods	101.01	Carroll PLC	Total next generation parallelism
16	eye	Beauty	119.97	Wiggins-Holland	Up-sized tertiary orchestration
17	under	Toys	64.63	Carr Inc	Devolved motivating knowledgebase
18	son	Electronics	36.85	Burke Group	Horizontal secondary intranet
19	forward	Home Goods	122.43	Jenkins-Cole	Object-based system-worthy extranet
20	prove	Toys	99.68	Hoffman, Meadows and Bishop	Multi-layered reciprocal infrastructure
21	similar	Toys	33.64	Blake Ltd	Streamlined solution-oriented projection
22	else	Home Goods	112.69	Hernandez-Alvarez	Persevering multimedia budgetary management
23	husband	Toys	14.22	Williams, Lawrence and Ruiz	Distributed bandwidth-monitored knowledge user
24	ok	Toys	154.60	Robinson-Munoz	Innovative global Local Area Network
25	window	Electronics	188.54	Baker-Jones	Business-focused high-level frame
26	produce	Groceries	120.43	Cameron Group	Ameliorated background hub
27	heavy	Beauty	188.06	Gonzalez, Johnson and Hood	Triple-buffered even-keeled conglomeration
28	machine	Home Goods	34.35	Schroeder-Maxwell	Public-key national installation
29	keep	Home Goods	46.14	Murray Ltd	Advanced bandwidth-monitored frame
30	become	Toys	56.86	Hall-Clarke	Re-engineered global model
31	best	Electronics	41.19	Yang-Henry	Devolved asymmetric protocol
32	strategy	Toys	92.50	Hall-Mercer	Compatible global archive
33	will	Clothing	122.74	Hughes-Guzman	Face-to-face explicit core
34	necessary	Beauty	176.28	Smith-Keller	User-friendly methodical complexity
35	fine	Toys	14.30	Pollard-Tran	Streamlined logistical initiative
36	natural	Groceries	156.30	Miller PLC	Object-based heuristic infrastructure
37	sell	Clothing	66.95	Baker, Richards and Tyler	User-centric zero tolerance project
38	enjoy	Clothing	165.67	Banks LLC	Cross-platform grid-enabled attitude
39	style	Groceries	92.64	Fletcher Group	Triple-buffered zero tolerance encryption
40	information	Home Goods	47.76	Lawson-Stevenson	Ameliorated contextually-based challenge
41	amount	Groceries	136.14	Terry PLC	Grass-roots maximized archive
42	officer	Home Goods	72.61	Cohen and Sons	Distributed human-resource analyzer
43	religious	Clothing	189.28	Castro-Wright	Triple-buffered context-sensitive moratorium
44	citizen	Clothing	178.21	Rodriguez, Stanton and Ellis	Polarized bi-directional customer loyalty
45	thought	Clothing	193.14	Petty-Leon	Mandatory upward-trending monitoring
46	executive	Home Goods	110.44	Snyder-Miller	Organized even-keeled open system
47	during	Toys	165.83	Montoya, Hayes and Carter	Up-sized executive conglomeration
48	doctor	Beauty	149.24	Fuller-Smith	Seamless responsive hierarchy
49	be	Toys	18.82	Miller, Powell and Dalton	Monitored directional approach
50	minute	Electronics	74.95	Rivera Inc	Assimilated multimedia moratorium
51	sign	Toys	35.74	Hernandez LLC	Front-line interactive ability
52	door	Home Goods	120.24	Marshall, Hill and Simon	Up-sized holistic portal
53	could	Home Goods	195.18	Jimenez Ltd	Distributed asynchronous moderator
54	matter	Home Goods	102.85	Mullins-Burnett	Progressive modular forecast
55	four	Clothing	76.28	Wallace Ltd	Enterprise-wide client-driven utilization
56	tonight	Clothing	141.28	Patterson, McMahon and Stone	Quality-focused discrete paradigm
57	entire	Groceries	179.03	Rivera, Gonzales and Hernandez	Reactive responsive superstructure
58	energy	Groceries	195.58	Greer Inc	Configurable heuristic architecture
59	art	Beauty	145.33	Smith-Herring	Triple-buffered uniform focus group
60	science	Home Goods	10.30	Barron-Allen	Inverse dedicated website
61	save	Groceries	77.87	White Inc	Digitized 24hour Graphical User Interface
62	care	Beauty	152.45	Flowers PLC	Self-enabling homogeneous collaboration
63	officer	Beauty	175.36	Curry, Carpenter and Richards	Reverse-engineered global strategy
64	player	Clothing	105.82	Kennedy-Nelson	Integrated didactic architecture
65	choose	Clothing	119.69	Wright Group	Cross-platform local adapter
66	already	Beauty	74.31	Ward, Gonzales and Cooley	Integrated homogeneous workforce
67	attack	Clothing	109.93	Harmon, Barnes and Burnett	Enterprise-wide zero administration workforce
68	majority	Home Goods	126.91	Cantrell-Hardin	Pre-emptive interactive system engine
69	sea	Groceries	165.08	Smith, Lewis and Harris	Fundamental optimal policy
70	environmental	Clothing	164.38	Bradley, Cannon and Horn	Inverse neutral framework
71	woman	Groceries	36.26	Fischer, Lewis and Gray	Enhanced multi-state strategy
72	road	Clothing	46.78	Bailey, Anderson and Cole	Adaptive fresh-thinking moderator
73	guess	Home Goods	20.12	Sullivan-Nguyen	Upgradable dedicated challenge
74	support	Clothing	41.23	Donovan-Ford	Exclusive modular attitude
75	rock	Groceries	82.84	Johnson LLC	Enterprise-wide bi-directional customer loyalty
76	whom	Home Goods	94.81	Guerra-Conley	Devolved composite groupware
77	ago	Clothing	136.18	Walker, Ferrell and Lambert	Multi-layered secondary budgetary management

78	magazine	Clothing	51.17	Fox Inc	Balanced uniform orchestration
79	indeed	Electronics	108.93	George, Day and Walker	Operative solution-oriented matrix
80	admit	Home Goods	147.08	Medina, Mckay and Collins	Front-line 4thgeneration pricing structure
81	ever	Beauty	21.29	Brown-Mills	Synergized background system engine
82	explain	Clothing	50.10	Vincent-Howe	Cross-group bi-directional Graphic Interface
83	must	Electronics	12.11	Kramer Inc	User-centric regional matrix
84	country	Clothing	118.10	James-Moore	Total explicit data-warehouse
85	send	Toys	46.83	Stewart, Dudley and Anderson	Managed systematic adapter
86	white	Toys	112.59	Morales, Camacho and Price	Cross-platform non-volatile interface
87	author	Electronics	145.44	Terry-Bishop	Open-architected regional structure
88	our	Electronics	129.03	Payne-Miller	Pre-emptive regional knowledge user
89	measure	Groceries	140.09	Patterson, White and Ponce	Optional secondary monitoring
90	too	Clothing	34.73	Potter Inc	Cross-platform cohesive software
91	current	Groceries	179.29	Goodman LLC	Persistent zero-defect strategy
92	just	Clothing	93.34	Lopez and Sons	Ameliorated disintermediate artificial intelligence
93	southern	Clothing	112.32	Lawson Inc	Total bandwidth-monitored moratorium
94	land	Toys	120.41	Allen, Brown and Osborne	Synergistic coherent architecture
95	force	Clothing	169.53	Montgomery, Scott and Gray	Automated user-facing hierarchy
96	seem	Groceries	30.32	Jones Inc	Synergistic optimal migration
97	read	Home Goods	195.71	May, Delgado and Vasquez	Enterprise-wide client-driven functionalities
98	source	Home Goods	49.84	Bennett, Lopez and Watson	Profit-focused empowering product
99	seat	Electronics	46.76	Wallace-Bennett	Visionary foreground forecast
100	ground	Clothing	69.93	Howard, Keller and Stafford	Innovative heuristic system engine

100 rows in set (0.00 sec)

So, we can actually populate all the 3 tables with fake data and can simulate transaction by writing records in the transactions table. Also, we can do query, subquery on the given tables.

```
mysql> select category, count(product_name) as totcount from products group by category order by totcount desc;
```

category	totcount
Clothing	23
Toys	22
Home Goods	20
Groceries	14
Electronics	11
Beauty	10

6 rows in set (0.00 sec)

Or find the second highest total count from above

```
mysql> select category, count(product_name) as totcount from products group by category order by totcount desc limit 1 offset 1;
```

category	totcount
Toys	22

1 row in set (0.00 sec)

**Cool!**

Contribute to the author \$5, [paypal.me/somnathpaulchoudhury](https://www.paypal.me/somnathpaulchoudhury)