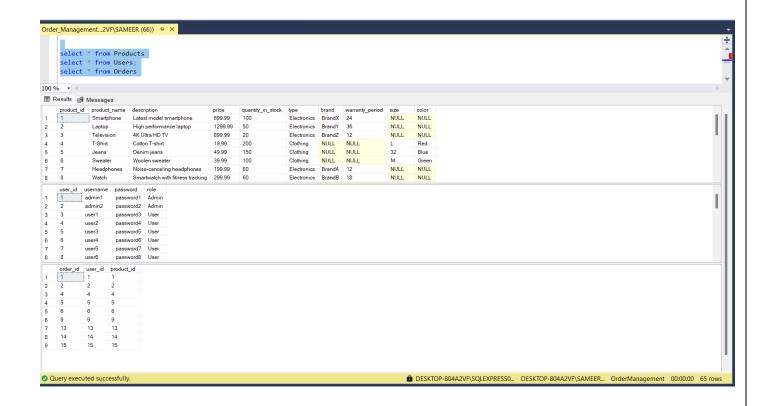


Create SQL Schema from the product and user class, use the class attributes for table column names:



# **Tasks**

1. Create a base class called Product with the following attributes - productId (int), productName (String), description (String), price (double), quantityInStock (int), type (String) [Electronics/Clothing].

```
# 1. Create a base class called Product with the following attributes:
class Product:

def __init__(self, productName, description, price, quantityInStock, type, productId=None):
    self.productId = productId
    self.productName = productName
    self.description = description
    self.price = price
    self.quantityInStock = quantityInStock
    self.type = type
```

2. Implement constructors, getters, and setters for the Product class.

```
# Getters
         def get_product_id(self) -> int:
            return self.productId
         def get_product_name(self) -> str:
           return self.productName
         def get_description(self) -> str:
            return self.description
         def get_price(self) -> float:
            return self.price
27
         def get_quantity_in_stock(self) -> int:
         return self.quantityInStock
         def get_type(self) -> str:
            return self.type
         # Setters
         def set_product_name(self, productName: str):
             self.productName = productName
         def set_description(self, description: str):
            self.description = description
         def set_price(self, price: float):
             self.price = price
         def set_quantity_in_stock(self, quantityInStock: int):
             self.quantityInStock = quantityInStock
```

3. Create a subclass Electronics that inherits from Product. Add attributes specific to electronics products, such as - brand (String) warrantyPeriod (int)

```
entity > 🕏 electronics.py > 😭 Electronics
      from entity.product import Product
      class Electronics(Product):
         def __init__(self, productName, description, price, quantityInStock, brand, model, warranty, power_usage):
              super().__init__(self, productName, description, price,
                              quantityInStock, "Electronics") # Call the base class constructor
              self.brand = brand # Brand of the electronic product
              self.warranty = warranty # Warranty period (in months or years)
         def getBrand(self):
             return self.brand
         def getWarrantyPeriod(self):
             return self.warrantyPeriod
         # Setters
         def setBrand(self, brand):
              self.brand = brand
         def setWarrantyPeriod(self, warrantyPeriod):
              self.warrantyPeriod = warrantyPeriod
```

4. Create a subclass Clothing that also inherits from Product. Add attributes specific to clothing products, such as - size (String), color (String)

```
entity > 🏓 clothing.py > 😉 Clothing > 😭 getSize
      from entity.product import Product
      class Clothing(Product):
          def __init__(self, productName, description, price, quantityInStock, size, color):
              super().__init__(self, productName, description, price, quantityInStock, "Clothing")
              self.size = size # Size of the clothing item (e.g., S, M, L, XL)
              self.color = color
          # Getters
15
          def getSize(self):
             return self.size
          def getColor(self):
              return self.color
          def setSize(self, size):
              self.size = size
          def setColor(self, color):
              self.color = color
```

5. Create a User class with attributes - userId (int), username (String), password (String), role (String) - "Admin" or "User

6. Define an interface/abstract class named IOrderManagementRepository

```
# ABSTRACT CLASS
# 6. Define an interface/abstract class named 10rderManagementRepository

from abc import ABC, abstractmethod

class IOrderManagementRepository(ABC):

@abstractmethod
def create_order(self, user, products):

pass

@abstractmethod
def cancel_order(self, userId, orderId):
pass

@abstractmethod
def create_product(self, user, product):
pass

@abstractmethod
def create_user(self, user):
pass

@abstractmethod
def create_user(self, user):
pass

@abstractmethod
def get_all_products(self):
pass

@abstractmethod
def get_all_products(self, user):
pass

@abstractmethod
def get_order_by_user(self, user):
pass
```

7. Implement the IOrderManagementRepository interface/abstractclass

Code in: Order\_Process\_class (Implementation)

8. Create DBUtil class and add the following method ( DATABASE CONNECTION )

```
# 8. Create DBUtil class
       class DBUtil:
           @staticmethod
           def getDBConn() -> pyodbc.Connection:
               connection_string = (
                  r'Driver={SQL Server};'
r'Server=DESKTOP-804A2VF\SQLEXPRESS09;'
                connection = pyodbc.connect(connection_string)
print("Connected Successfully")
                   print("Connection failed: {}".format(e))
       db_connection = DBUtil.getDBConn()
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Connected Successfully
Connected Successfully
 Order Management System
2. Create Product
3. Cancel Order
4. Get All Products
5. Get Orders by User
6. Exit
Enter your choice:

* History restore
PS \ C: \ Users SAMEER \ One Drive \ Desktop \ HExa \ Coding Challenges \ Order Management > python \ util/db\_connection\_util.py
\label{lem:connected} Connected Successfully $$PS C:\Users\SAMEER\OneDrive\Desktop\HExa\CodingChallenges\OrderManagement> $$[]
```

9. Create OrderManagement main class and perform following operation:

# CreateUser

#### Before:

	user_id	username	password	role
29	33	sameer	pal@123	Admin
30	34	sam	eer	Admin
31	35	sartha	sas#234	Admin
32	37	sdf	wer	Admin
33	38	s	f	Admin

# **User Input:**

# Order Management System

- 1. Create User
- 2. Create Product
- 3. Cancel Order
- 4. Get All Products
- 5. Get Orders by User
- 6. Exit

Enter your choice: 1
Enter username: sarthak
Enter password: sharma

Enter role (Admin/User): Admin User sarthak created successfully. User sarthak created with role Admin

#### After:

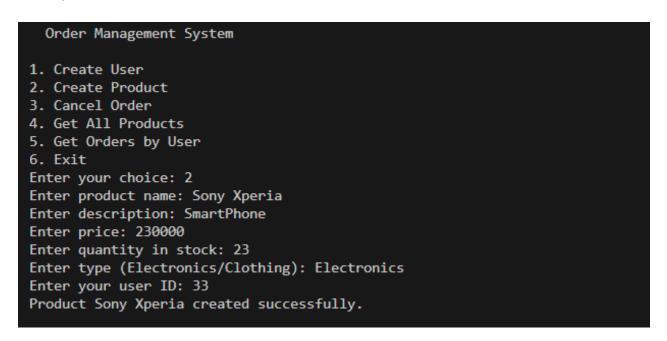
	user_id	username	password	role
30	34	sam	eer	Admin
31	35	sartha	sas#234	Admin
32	37	sdf	wer	Admin
33	38	s	f	Admin
34	39	sarthak	sharma	Admin

### CreateProduct

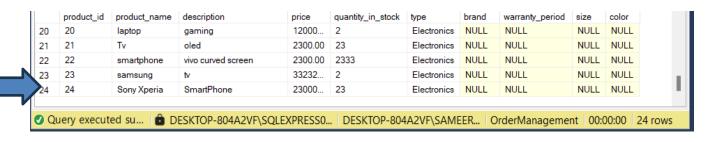
#### Before:

	product_id	product_name	description	price	quantity_in_stock	type	brand	warranty_period	size	color
19	19	phone	samsusg A23	34000	45	Electronics	NULL	NULL	NULL	NULL
20	20	laptop	gaming	12000	2	Electronics	NULL	NULL	NULL	NULL
21	21	Tv	oled	2300.00	23	Electronics	NULL	NULL	NULL	NULL
22	22	smartphone	vivo curved screen	2300.00	2333	Electronics	NULL	NULL	NULL	NULL
23	23	samsung	tv	33232	2	Electronics	NULL	NULL	NULL	NULL

#### **User Input:**



#### After:



# **CancelOrder:**

# When Order ID doesn't Exist:

Enter order ID: 9

Order ID 9 not found for User ID 1.

# Order Management System 1. Create User 2. Create Product 3. Cancel Order 4. Get All Products 5. Get Orders by User 6. Exit Enter your choice: 3 Enter user ID: 1



	order_id	user_id	product_id
5	6	6	6
6	9	9	9
7	13	13	13
8	14	14	14
9	15	15	15

# **User Input:**

```
Order Management System

1. Create User
2. Create Product
3. Cancel Order
4. Get All Products
5. Get Orders by User
6. Exit
Enter your choice: 3
Enter user ID: 15
Enter order ID: 15
Order 15 canceled for User ID 15.
```

#### After:

	order_id	user_id	product_id
4	5	5	5
5	6	6	6
6	9	9	9
7	13	13	13
8	14	14	14

#### **GetAllProducts**

```
Order Management System

1. Create User
2. Create Product
3. Cancel Order
4. Get All Products
5. Get Orders by User
6. Exit
Enter your choice: 3
Enter user ID: 15
(6, 'Sweater', 'Woolen sweater', Decimal('39.99'), 180, 'Clothing', None, Non
```

# **GetOrderbyUser:**

```
Order Management System

1. Create User
2. Create Product
3. Cancel Order
4. Get All Products
5. Get Orders by User
6. Exit
Enter your choice: 5
Enter user ID: 4
(4, 'T-Shirt', 'Cotton T-shirt', Decimal('19.99'), 4, 'user2', 'User')
```

#### Join - Product, Users, Orders

	product_ic	d product	_name de	scription		price	quantity_in_stock	type	brand	warranty_period	size	color
1	1	Smartp	hone L	test mode	smartphone	699.99	100	Electronics	BrandX	24	NULL	NUL
2	2	Laptop	Н	gh perfori	nance laptop	1299.99	50	Electronics	BrandY	36	NULL	NUL
3	3	Televisi	on 4	(Ultra HD	TV	899.99	20	Electronics	BrandZ	12	NULL	NUL
	4	T-Shirt	С	otton T-shi	t	19.99	200	Clothing	NULL	NULL	L	Red
5	5	Jeans	D	enim jean	3	49.99	150	Clothing	NULL	NULL	32	Blue
6	6	Sweate	r V	oolen swe	ater	39.99	100	Clothing	NULL	NULL	М	Gre
7	7	Headph	nones N	oise-canc	ling headp	199.99	80	Electronics	BrandA	12	NULL	NUI
8	8	Watch	S	martwatch	with fitness	299.99	60	Electronics	BrandB	18	NULL	NUI
9	9	Shirt	F	rmal shirt		29.99	100	Clothing	NULL	NULL	М	Whi
10	10	Jacket	L	ather jack	et	129.99	50	Clothing	NULL	NULL	L	Blac
11_	11	Microw	ave C	onvection	microwave	99.99	40	Electronics	BrandC	24	NULL	NU
1 2 3	1 2 3 4	admin1 admin2 user1 user2	password password password password	2 Admi 3 User								
2	2	admin2 user1	password	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8	2 3 4 5 6 7 8	admin2 user1 user2 user3 user4 user5 user6 user7	password password password password password password password	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9	2 3 4 5 6 7 8 9	admin2 user1 user2 user3 user4 user5 user6 user7 user8	password password password password password password password password	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9	2 3 4 5 6 7 8 9 10 11 order_id	admin2 user1 user2 user3 user4 user5 user6 user7 user8 user8	password password password password password password password password password password	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9 10	2 3 4 5 6 7 8 9 10 11 order_id	admin2 user1 user2 user3 user4 user5 user6 user7 user8 user_id 1 2	password password password password password password password password password password password password password password	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9 10	2 3 4 5 6 7 8 9 10 11 order_id	admin2 user1 user2 user3 user4 user5 user6 user7 user8 user_id 1 2 4	password pas	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9 10 11 2 3 4	2 3 4 5 6 7 8 9 10 11 order_id 1 2 4 5	admin2 user1 user2 user3 user4 user5 user6 user7 user8 user_id 1 2 4 5	password pas	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9 10 1 2 3 4 5	2 3 4 5 6 7 8 9 10 11 order_id 1 2 4 5 6	admin2 user1 user2 user3 user4 user5 user6 user7 user8 user_id 1 2 4 5 6	password pas	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								
2 3 4 5 6 7 8 9 10 11 2 3 4	2 3 4 5 6 7 8 9 10 11 order_id 1 2 4 5	admin2 user1 user2 user3 user4 user5 user6 user7 user8 user_id 1 2 4 5	password pas	2 Admi 3 User 4 User 5 User 6 User 7 User 8 User 9 User								

