

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського» ФАКУЛЬТЕТ  
ПРИКЛАДНОЇ МАТЕМАТИКИ  
**Кафедра системного програмування та спеціалізованих комп'ютерних систем**

**РГР**

*“Проектування бази даних та ознайомлення з базовими операціями СУБД  
PostgreSQL”*

Виконав:

КВ-21 Мануйлов Денис

Варіант - Інвентаризація складського обліку

Посилання на GitHub - <https://github.com/LoPHarp/RGR.git>

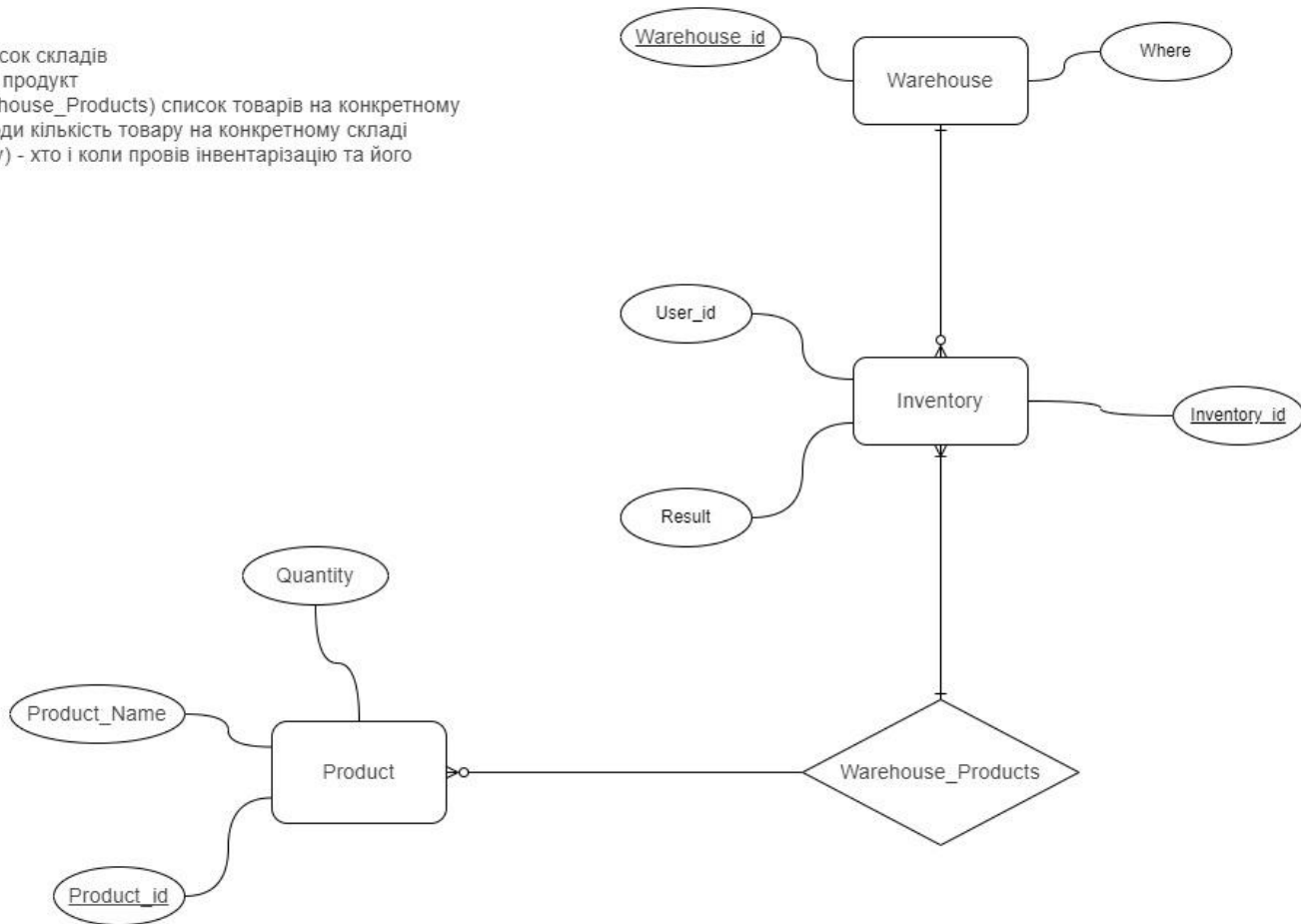
Контакти телеграм - <https://t.me/LoPHarp> або +380 96 038 51 97

### Пункт №1:

## БД "Інвентаризація складського обліку"

### Сутності:

1. Склад (Warehouse) - список складів
2. Товар (Product) - певний продукт
3. Товари\_на\_складі (Warehouse\_Products) список товарів на конкретному складі, можна включити сюди кількість товару на конкретному складі
4. Інвентаризація (Inventory) - хто і коли провів інвентаризацію та його висновок



### Пункт №2:

#### Сутності у таблиці:

Warehouse -> Warehouse

Inventory -> Inventory

Warehouse\_Products -> Warehouse\_Products

Product -> Product

#### Функціональні залежності у вигляді A-->B

##### 1 Warehouse:

Warehouse\_id → Where

##### 2 Inventory:

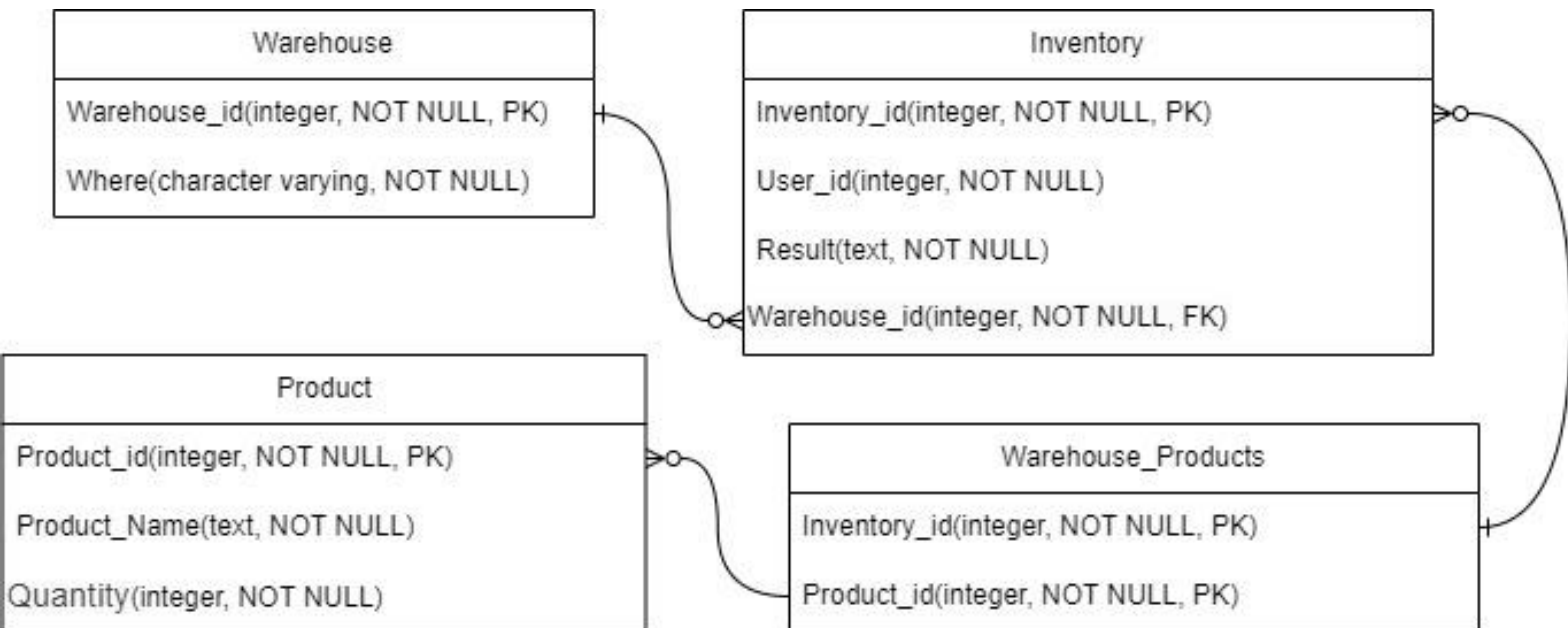
Inventory\_id → User\_id, Result, Warehouse\_id

##### 3 Product:

Product\_id → Product\_Name, Quantity

#### 4 Warehouse\_Products (зіставний ключ):

(Inventory\_id, Product\_id)→Quantity



Select a table or other function:

1. Warehouse
  2. Inventory
  3. Product
  4. Warehouse\_Products
  5. Quit
  6. Clear all tables
  7. Inventory Count by Warehouse
  8. Products with Quantity
  9. Warehouse with Most Inventories
- Enter your choice:

Menu:

1. Add line
  2. View line
  3. Update line
  4. Delete line
  5. Quit in menu
  6. Generate random strings
- Enter your choice:

1/2/3/4(Warehouse/Inventory/Product/Warehouse\_Product) – При виборі якоїсь з цих функцій перекине на наступну меню, яка пов'язана з обробкою таблиць.  
5(Quit) –

6(Clear all tables) – Зносить усі таблиці та завершує програму(користувався при тестах, вирішив залишити)

7(Inventory Count by Warehouse) - Скільки інвентаризацій було проведено на кожному складі

8(Products with Quantity) - Які продукти мають кількість більше заданого значення і прив'язані до складів

9(Warehouse with Most Inventories) - Склад із найбільшою кількістю інвентаризацій

Мова – python

Використанні бібліотеки:

`import psycorg2`

`import random`

`import time`

(останні 2 для налагодження певних випадкових заповнень таблиць, наприклад було додано у таблиці Inventory, стовпці Result з певним шансом генерується “В процесі” чи “Очікується”)

#### Приклад операції вилучення запису батьківської таблиці

Menu:

1. Add line
2. View line
3. Update line
4. Delete line
5. Quit in menu
6. Generate random strings

Enter your choice: 1

Enter Where: sity1

line added successfully!

Menu:

1. Add line
2. View line
3. Update line
4. Delete line
5. Quit in menu
6. Generate random strings

Enter your choice: 2

Lines:

Warehouse\_id: 1, Where: sity1

Select a table or other function:

1. Warehouse
2. Inventory
3. Product
4. Warehouse\_Products
5. Quit
6. Clear all tables
7. Inventory Count by Warehouse
8. Products with Quantity
9. Warehouse with Most Inventories

Enter your choice: 2

Menu:

1. Add line
2. View line
3. Update line
4. Delete line
5. Quit in menu
6. Generate random strings

Enter your choice: 1

Enter User\_id: 1

Enter Result: True

Enter Warehouse\_id: 1

line added successfully!

Select a table or other function:

1. Warehouse
  2. Inventory
  3. Product
  4. Warehouse\_Products
  5. Quit
  6. Clear all tables
  7. Inventory Count by Warehouse
  8. Products with Quantity
  9. Warehouse with Most Inventories
- Enter your choice: 1

Menu:

1. Add line
  2. View line
  3. Update line
  4. Delete line
  5. Quit in menu
  6. Generate random strings
- Enter your choice: 4
- Enter line ID: 1
- line deleted successfully!

Select a table or other function:

1. Warehouse
  2. Inventory
  3. Product
  4. Warehouse\_Products
  5. Quit
  6. Clear all tables
  7. Inventory Count by Warehouse
  8. Products with Quantity
  9. Warehouse with Most Inventories
- Enter your choice: 2

Menu:

1. Add line
  2. View line
  3. Update line
  4. Delete line
  5. Quit in menu
  6. Generate random strings
- Enter your choice: 2
- Lines:

Menu:

1. Add line

*Приклад операції вставки запису в дочірню таблицю, якщо в батьківській таблиці немає відповідного запису.*

Select a table or other function:

1. Warehouse
2. Inventory
3. Product
4. Warehouse\_Products
5. Quit
6. Clear all tables
7. Inventory Count by Warehouse
8. Products with Quantity
9. Warehouse with Most Inventories

Enter your choice: 2

Menu:

1. Add line
2. View line
3. Update line
4. Delete line
5. Quit in menu
6. Generate random strings

Enter your choice: 1

Enter User\_id: 2

Enter Result: False

Enter Warehouse\_id: 1

Error: Warehouse\_id 1 does not exist.

копії SQL-запитів, що ілюструють генерацію при визначених вхідних параметрах.

QueryQuery History

Scratch Pad x

1 INSERT INTO Warehouse ("Where")

2 SELECT 'Location' || trunc(1 + random()\*100)::int

3 FROM generate\_series(1, 100);

Data OutputMessagesNotifications

INSERT @ 100

Query returned successfully in 62 msec.

	warehouse_id [PK] integer	Where character varying
61	61	Location20
62	62	Location33
63	63	Location53
64	64	Location22
65	65	Location93
66	66	Location33
67	67	Location100
68	68	Location16
69	69	Location89
70	70	Location43
71	71	Location74
72	72	Location25
73	73	Location13
74	74	Location71
75	75	Location74
76	76	Location98
77	77	Location96
78	78	Location10
79	79	Location95
Total rows: 200		Query complete 00:00:00.260

Warehouse\_id: 198, Where: Location86  
Warehouse\_id: 199, Where: Location30  
Warehouse\_id: 200, Where: Location51

Menu:  
1. Add line  
2. View line  
3. Update line  
4. Delete line  
5. Quit in menu  
6. Generate random strings  
Enter your choice:

Menu:  
1. Add line  
2. View line  
3. Update line  
4. Delete line  
5. Quit in menu  
6. Generate random strings  
Enter your choice: 6  
Enter rows for Warehouse: 10  
10 rows added to Warehouse.

Warehouse\_id: 201, Where: location92  
Warehouse\_id: 202, Where: location32  
Warehouse\_id: 203, Where: location5  
Warehouse\_id: 204, Where: location89  
Warehouse\_id: 205, Where: location79  
Warehouse\_id: 206, Where: location81  
Warehouse\_id: 207, Where: location37  
Warehouse\_id: 208, Where: location17  
Warehouse\_id: 209, Where: location61  
Warehouse\_id: 210, Where: location85

Menu:  
1. Add line  
2. View line  
3. Update line  
4. Delete line  
5. Quit in menu  
6. Generate random strings  
Enter your choice: |

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

```
WITH WarehouseSample AS (  
    SELECT Warehouse_id  
    FROM Warehouse  
    ORDER BY RANDOM()  
    LIMIT 100  
),  
RandomData AS (  
    SELECT  
        FLOOR(RANDOM() * 300 + 1)::INT AS User_id,  
        CASE  
            WHEN RANDOM() < 0.45 THEN 'True'  
            WHEN RANDOM() < 0.90 THEN 'False'  
            WHEN RANDOM() < 0.92 THEN 'Unsuccessful'  
            WHEN RANDOM() < 0.95 THEN 'In process'  
            ELSE 'Planned'  
        END AS Result,  
        ROW_NUMBER() OVER () AS RowNum  
    FROM GENERATE_SERIES(1, 100) AS s  
)  
INSERT INTO Inventory (User_id, Result, Warehouse_id)  
SELECT  
    rd.User_id,  
    rd.Result,  
    ws.Warehouse_id  
FROM RandomData rd  
JOIN WarehouseSample ws  
ON rd.RowNum = ws.Warehouse_id % 100 + 1;
```

Data Output

Messages

Notifications

SQL

Showing rows: 1 to 20 of 1

	inventory_id [PK] integer	user_id integer	result text	warehouse_id integer
1	1	210	False	10
2	2	55	False	1
3	3	176	True	2
4	4	70	True	3
5	5	74	False	4
6	6	297	False	5
7	7	45	True	6
8	8	292	False	7
9	9	133	False	8
10	10	177	False	9
11	11	129	False	1
12	12	276	True	2
13	13	120	True	3
14	14	195	False	4
15	15	186	False	5
16	16	210	False	6

Total rows: 20

Query complete 00:00:00.082

Menu:

1. Add line

2. View line

3. Update line

4. Delete line

5. Quit in menu

6. Generate random strings

Enter your choice: 6

Enter rows for Inventory: 10

10 rows added to Inventory.

Menu:

1. Add line

2. View line

3. Update line

4. Delete line

5. Quit in menu

6. Generate random strings

Enter your choice: 2

Lines:

Inventory\_id: 1, User\_id: 210, Result: False, Warehouse\_id: 10

Inventory\_id: 2, User\_id: 55, Result: False, Warehouse\_id: 1

Inventory\_id: 3, User\_id: 176, Result: True, Warehouse\_id: 2

Inventory\_id: 4, User\_id: 70, Result: True, Warehouse\_id: 3

Inventory\_id: 5, User\_id: 74, Result: False, Warehouse\_id: 4

Inventory\_id: 6, User\_id: 297, Result: False, Warehouse\_id: 5

Inventory\_id: 7, User\_id: 45, Result: True, Warehouse\_id: 6

Inventory\_id: 8, User\_id: 292, Result: False, Warehouse\_id: 7

Inventory\_id: 9, User\_id: 133, Result: False, Warehouse\_id: 8

Inventory\_id: 10, User\_id: 177, Result: False, Warehouse\_id: 9

**У інших сутностях, була використана проста генерація чисел, наприклад при генерації назв продуктів: *Product* + “Випадкове число”.**



Ілюстрації уведення пошукового запиту та результатів виконання запитів

Приклад пошуку за 1 атрибутом:

QueryQuery History

Scratch Pad X

1 SELECT w."Where", w.Warehouse\_id, i.Inventory\_id, i.User\_id,  
2 i.Result, p.Product\_id, p.Product\_Name, p.Quantity  
3 FROM public.Warehouse w  
4 JOIN public.Inventory i ON w.Warehouse\_id = i.Warehouse\_id  
5 JOIN public.Warehouse\_Products wp ON i.Inventory\_id = wp.Inventory\_id  
6 JOIN public.Product p ON wp.Product\_id = p.Product\_id  
7 WHERE  
8 -- ('Location57'::text IS NULL OR w."Where" LIKE 'Location57'  
9 ('9'::int IS NULL OR i.Inventory\_id = '9')  
10 -- AND ('92'::int IS NULL OR i.User\_id = '92')  
11 -- AND ('True'::text IS NULL OR i.Result ILIKE 'True')  
12 -- AND ('4'::int IS NULL OR p.Product\_id = '4')  
13 -- AND (''::int IS NULL OR w.Warehouse\_id = '')  
14 -- AND (''::text IS NULL OR p.Product\_Name = '')  
15 -- AND (\$8::int IS NULL OR p.Quantity >= \$8)  
16 -- AND (\$9::int IS NULL OR p.Quantity <= \$9);

Data OutputMessagesNotifications

Showing rows: 1 to 2Page No: 1 of 1

	Where character varying	warehouse_id integer	inventory_id integer	user_id integer	result text	product_id integer	product_name text	quantity integer
1	Location57	4	9	92	True	4	product757	820
2	Location57	4	9	92	True	389	product839	450

Приклад пошуку за декількома атрибутами:

QueryQuery History

Scratch Pad X

1 SELECT w."Where", w.Warehouse\_id, i.Inventory\_id, i.User\_id,  
2 i.Result, p.Product\_id, p.Product\_Name, p.Quantity  
3 FROM public.Warehouse w  
4 JOIN public.Inventory i ON w.Warehouse\_id = i.Warehouse\_id  
5 JOIN public.Warehouse\_Products wp ON i.Inventory\_id = wp.Inventory\_id  
6 JOIN public.Product p ON wp.Product\_id = p.Product\_id  
7 WHERE  
8 ('Location57'::text IS NULL OR w."Where" LIKE 'Location57')  
9 AND ('9'::int IS NULL OR i.Inventory\_id = '9')  
10 AND ('92'::int IS NULL OR i.User\_id = '92')  
11 AND ('True'::text IS NULL OR i.Result ILIKE 'True')  
12 AND ('4'::int IS NULL OR p.Product\_id = '4')  
13 -- AND (''::int IS NULL OR w.Warehouse\_id = '')  
14 -- AND (''::text IS NULL OR p.Product\_Name = '')  
15 -- AND (\$8::int IS NULL OR p.Quantity >= \$8)  
16 -- AND (\$9::int IS NULL OR p.Quantity <= \$9);

Data OutputMessagesNotifications

Showing rows: 1 to 1Page No: 1 of 1

	Where character varying	warehouse_id integer	inventory_id integer	user_id integer	result text	product_id integer	product_name text	quantity integer
1	Location57	4	9	92	True	4	product757	820

Теж саме, але тільки кодом:

```
7. Search Across All Entities
Enter your choice: 7
Enter search criteria (leave blank to skip):
Warehouse 'Where':
Inventory ID: 9
User ID:
Result (e.g., True, False, Planned):
Warehouse ID:
Product ID:
Product Name (partial match):
Min Quantity:
Max Quantity:

Search Results
-----
Where      | Warehouse ID | Inventory ID | User ID | Result | Product ID | Product Name | Quantity
-----
Location57 | 4             | 9            | 92      | True   | 4          | product757   | 820
Location57 | 4             | 9            | 92      | True   | 389        | product839   | 450
-----

Select a table or other function:
1. Warehouse
2. Inventory
3. Product
4. Warehouse_Products
5. Quit
6. Clear all tables
7. Search Across All Entities
Enter your choice: 7
Enter search criteria (leave blank to skip):
Warehouse 'Where': Location57
Inventory ID: 9
User ID: 92
Result (e.g., True, False, Planned): True
Warehouse ID: 4
Product ID: 4
Product Name (partial match): product757
Min Quantity: 555
Max Quantity: 999

Search Results
-----
Where      | Warehouse ID | Inventory ID | User ID | Result | Product ID | Product Name | Quantity
-----
Location57 | 4             | 9            | 92      | True   | 4          | product757   | 820
-----
```

*ілюстрації програмного коду модуля “Model”, згідно із шаблоном MVC*

```
1  import psycopg2
2  import random
3  import time
4
5  class Model: 2 usages  Aroxed *
6  >     def __init__(self):...
15
16 >     def create_tables(self):...
59
60 >     def add_line_Warehouse(self, Where):...
64
65 >     def add_line_Inventory(self, User_id, Result, Warehouse_id):...
73
74 >     def add_line_Product(self, Product_Name, Quantity):...
78
79 >     def add_line_Warehouse_Products(self, Inventory_id, Product_id):...
91
92 >     def get_all_line(self, choice_table):...
105
106
107 >     def update_line_Warehouse(self, Warehouse_id, Where):...
111
112 >     def update_line_Inventory(self, Inventory_id, User_id, Result, Warehouse_id):...
120
121 >     def update_line_Product(self, Product_id, Product_Name, Quantity):...
125
126 >     def update_line_Warehouse_Products(self, old_Inventory_id, old_Product_id, new_Inventory_id, new_Product_id):...
138
139 >     def delete_line(self, line_id, choice_table):...
150
151 >     def delete_line_Warehouse_Products(self, Inventory_id, Product_id):...
155
156 >     def bulk_insert_warehouse(self, count):...
161
162 >     def bulk_insert_inventory(self, count):...
178
179 >     def bulk_insert_product(self, count):...
186
187 >     def bulk_insert_warehouse_products(self, count):...
208
209 >     def drop_all_tables(self):...
221
222 >     def get_inventory_count_by_warehouse(self, result_filter):...
236
237 >     def get_products_with_quantity(self, min_quantity):...
251
252 >     def get_warehouse_with_most_inventories(self):...
267
```

## **1 Ініціалізація:**

- **`__init__`** — встановлює з'єднання з базою даних і викликає метод **`create_tables`** для створення таблиць.

## **2 Створення таблиць:**

- **`create_tables`** — створює таблиці: **`Warehouse`**, **`Inventory`**, **`Product`** та **`Warehouse_Products`**, якщо вони ще не існують.

## **3 Додавання записів:**

- **`add_line_Warehouse`**, **`add_line_Inventory`**, **`add_line_Product`**, **`add_line_Warehouse_Products`** — додають записи в відповідні таблиці, перевіряючи коректність введених даних.

## **4 Оновлення записів:**

- **`update_line_Warehouse`**, **`update_line_Inventory`**, **`update_line_Product`**, **`update_line_Warehouse_Products`** — оновлюють записи у відповідних таблицях з перевіркою існування записів.

## **5 Видалення записів:**

- **`delete_line`**, **`delete_line_Warehouse_Products`** — видаляють записи з таблиць.

## **6 Отримання даних:**

- **`get_all_line`** — повертає всі записи з вибраної таблиці.
- **`get_inventory_count_by_warehouse`**, **`get_products_with_quantity`**, **`get_warehouse_with_most_inventories`** — виконують запити для отримання аналітичної інформації (наприклад, кількість інвентаризацій на складах, продукти з мінімальною кількістю, склад із найбільшою кількістю інвентаризацій).

## **7 Масове додавання даних:**

- **`bulk_insert_warehouse`**, **`bulk_insert_inventory`**, **`bulk_insert_product`**, **`bulk_insert_warehouse_products`** — автоматично додають випадкові записи в таблиці.

## **8 Управління базою даних:**

- **`drop_all_tables`** — видаляє всі таблиці.