

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>

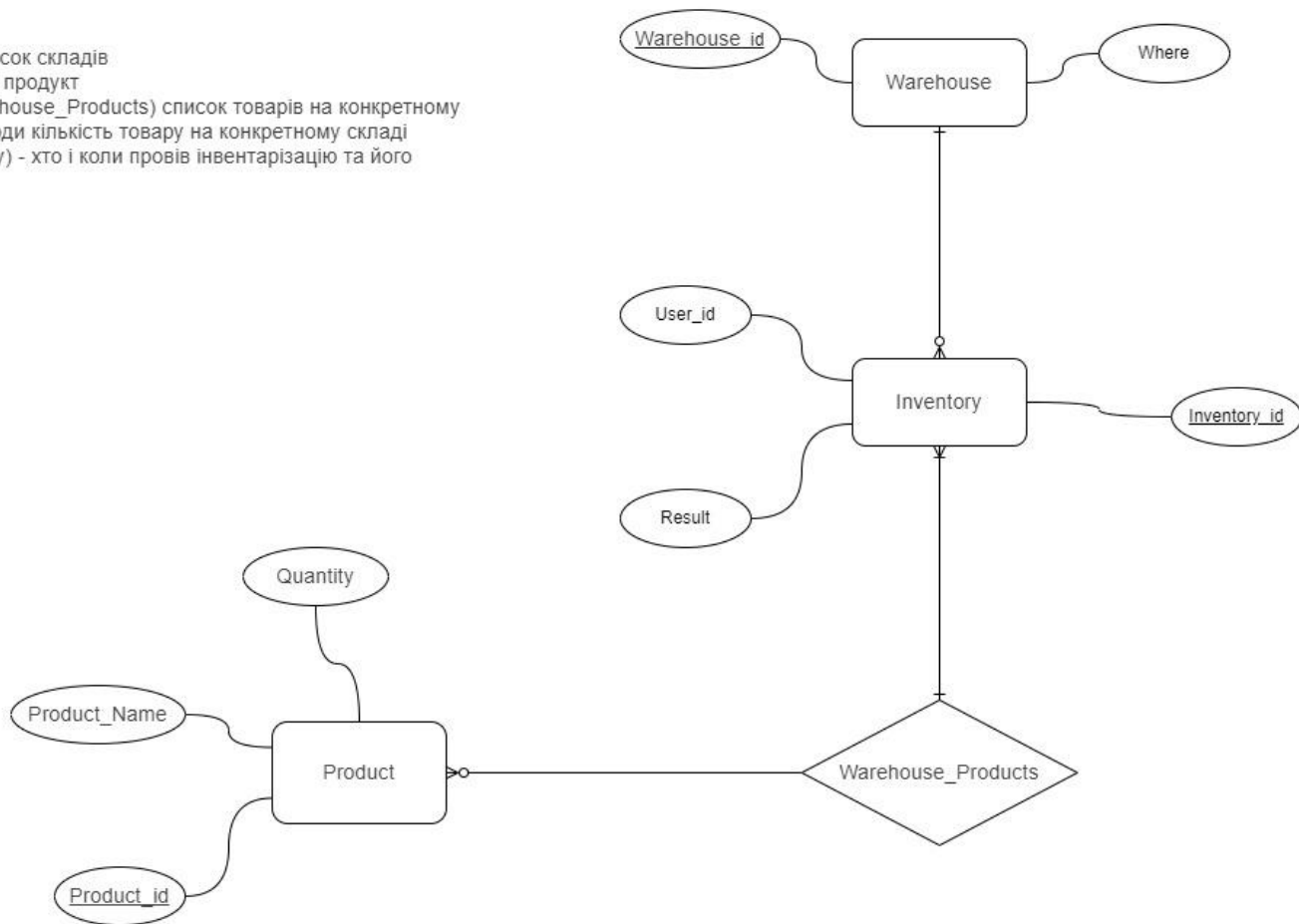
Контакти телеграм - <https://t.me/LoPHarp> або +380 96 038 51 97

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:

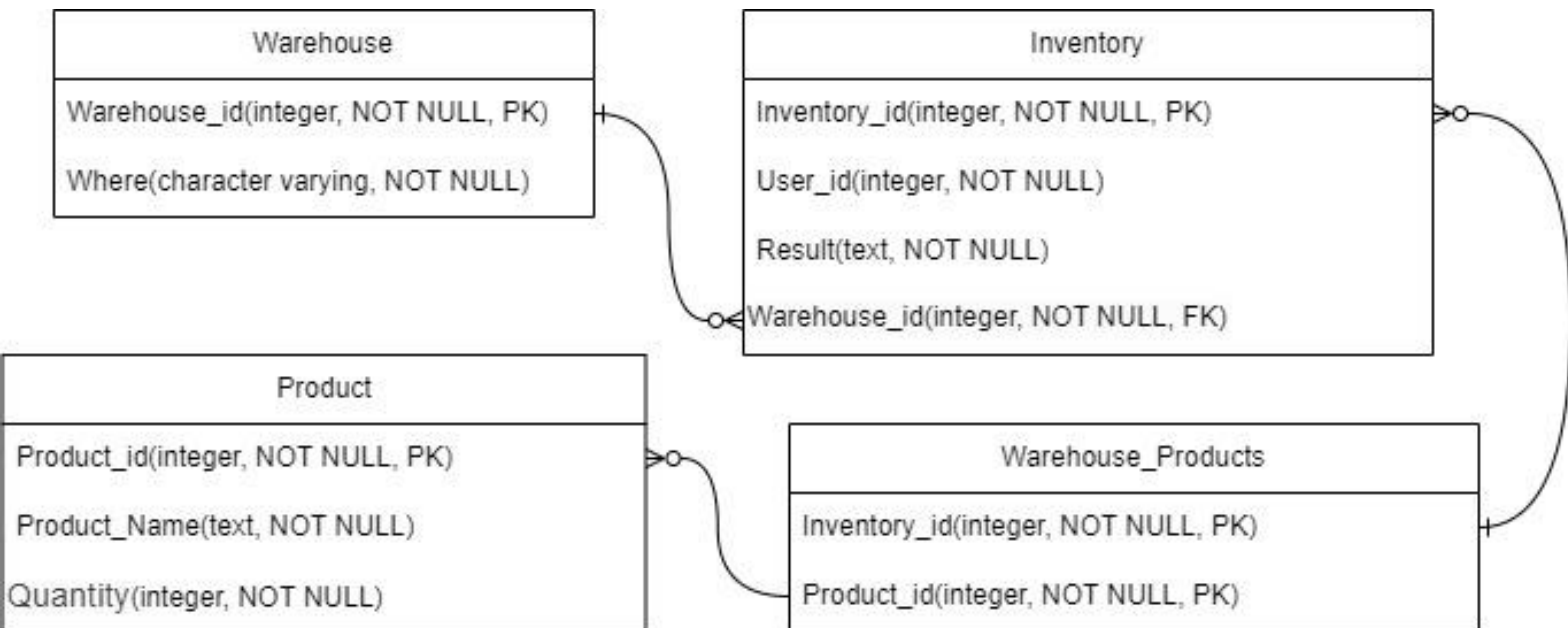
Inventory_id → User_id, Result, Warehouse_id

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 HistoryScratch Pad x

1 INSERT INTO Warehouse ("Where")
2 SELECT 'Location' || trunc(1 + random()*100)::int
3 FROM generate_series(1, 100);

Data Output Messages Notifications

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:
Enter rows for Warehouse: 10
10 rows added to Warehouse.

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

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: |

Усі інші на половину з коду python тому продемонструвати їх SQL запит я не можу

```
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: 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: 220, Result: False, Warehouse_id: 171
Inventory_id: 2, User_id: 172, Result: True, Warehouse_id: 48
Inventory_id: 3, User_id: 90, Result: False, Warehouse_id: 93
Inventory_id: 4, User_id: 184, Result: False, Warehouse_id: 106
Inventory_id: 5, User_id: 292, Result: Planned, Warehouse_id: 171
Inventory_id: 6, User_id: 138, Result: True, Warehouse_id: 125
Inventory_id: 7, User_id: 244, Result: True, Warehouse_id: 182
Inventory_id: 8, User_id: 81, Result: True, Warehouse_id: 54
Inventory_id: 9, User_id: 92, Result: True, Warehouse_id: 4
Inventory_id: 10, User_id: 193, Result: True, Warehouse_id: 109
```

```
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: 3
```

```
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 Product: 10
10 rows added to Product.
```

```
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:
Product_id: 1, Product_Name: product45, Quantity: 990
Product_id: 2, Product_Name: product201, Quantity: 493
Product_id: 3, Product_Name: product35, Quantity: 1047
Product_id: 4, Product_Name: product757, Quantity: 820
Product_id: 5, Product_Name: product219, Quantity: 906
Product_id: 6, Product_Name: product986, Quantity: 336
Product_id: 7, Product_Name: product460, Quantity: 1484
Product_id: 8, Product_Name: product794, Quantity: 1122
Product_id: 9, Product_Name: product937, Quantity: 296
Product_id: 10, Product_Name: product581, Quantity: 964
```

```
6. Clear all tables
7. Inventory Count by Warehouse
8. Products with Quantity
9. Warehouse with Most Inventories
Enter your choice: 4
```

```
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_Products: 10
10 rows added to Warehouse_Products.
```

```
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: 2, Product_id: 3
Inventory_id: 6, Product_id: 2
Inventory_id: 1, Product_id: 10
Inventory_id: 5, Product_id: 2
Inventory_id: 10, Product_id: 9
Inventory_id: 2, Product_id: 5
Inventory_id: 8, Product_id: 6
Inventory_id: 9, Product_id: 4
Inventory_id: 7, Product_id: 1
Inventory_id: 4, Product_id: 2
```


ілюстрації уведення пошукового запиту та результатів виконання запитів

```
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: 8
Enter minimum quantity: 200
Products with Quantity > 200
product581 | 1
product35 | 1
product219 | 1
product757 | 1
product201 | 3
product937 | 1
product45 | 1
product986 | 1
Query executed in 1.04 ms.
```

```
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: 7
Enter Result filter (e.g., True, False, Planned): True
Inventory Count by Warehouse (Filter: True)
Location100 | 1
Location2 | 1
Location23 | 1
Location37 | 1
Location57 | 1
Location90 | 1
Query executed in 0.00 ms.
```

Тут з SQL запитами те ж саме

що й при генерації рядків

```
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: 9
Warehouse with Most Inventories
Location63 | 2
Query executed in 1.03 ms.
```

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

```
1  import psycpg2
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.