**Wstęp teoretyczny**

Gospodarka magazynowa to obszar logistyki zajmujący się zarządzaniem przepływem towarów w przedsiębiorstwie. Obejmuje:

* Dostawy towarów do magazynu
* Sprzedaż i wydania z magazynu
* Zwroty od klientów
* Reklamacje i usuwanie towarów z obiegu
* Analizę zapasów i ruchów magazynowych

W systemach takich jak SAP, IFS czy inne ERP, te procesy są obsługiwane w modułach takich jak MM (Materials Management) i WM (Warehouse Management). My odwzorujemy je w uproszczonej wersji w Pythonie.

**Wymagania techniczne**

* Minimum Python 3.5
* SQLite3 (wbudowany)
* Biblioteki:
  + tkinter (GUI)
  + sqlite3 (baza danych)
  + matplotlib (wykresy)
  + pandas (analizy)

Instalacja brakujących bibliotek:

*pip install matplotlib pandas*

**Wprowadzenie**

SQLite to lekka, szybka i wbudowana baza danych SQL, która nie wymaga instalacji serwera ani konfiguracji użytkowników. Jest to najczęściej używana baza danych na świecie, obecna m.in. w przeglądarkach internetowych, aplikacjach mobilnych, a także w wielu systemach osadzonych (embedded).

**Cechy SQLite:**

* **Plikowa baza danych** – cała zawartość bazy zapisana jest w jednym pliku .db, co ułatwia przenoszenie i backup.
* **Brak konieczności instalowania serwera** – wszystko działa bezpośrednio z poziomu aplikacji (np. w Pythonie).
* **Wysoka wydajność** przy małych i średnich projektach.
* **Zgodność z SQL** – obsługuje większość standardowych zapytań SQL (SELECT, INSERT, JOIN, itd.).
* Idealna do **prototypowania, nauki oraz aplikacji lokalnych**, takich jak nasz projekt gospodarki magazynowej.

**SQLite + Python**

SQLite jest wbudowany w standardową bibliotekę Pythona poprzez moduł sqlite3, co pozwala na natychmiastowe rozpoczęcie pracy – bez instalowania zewnętrznych bibliotek.Przykład podstawowego połączenia z bazą danych w Pythonie pokazano w załączonych skryptach.

**Struktura systemu**

**Tabela Produkty**

* ProduktID – ID produktu
* Nazwa, Kategoria, Cena, Ilosc

**Tabela Klienci**

* KlientID, Imie, Nazwisko, Email

**Tabela Zamowienia**

* ZamowienieID, KlientID, ProduktID, Ilosc, DataZamowienia

**Tabela OperacjeMagazynowe**

* OperacjaID, ProduktID, TypOperacji, Ilosc, DataOperacji, Uwagi

**Cel ćwiczenia**

Celem zajęć jest dodanie do istniejącego systemu:

* Modułu zamówień klientów (baza danych + GUI),
* Możliwości realizacji zamówień tylko przy dostępności towaru,
* Śledzenia operacji sprzedażowych jako operacji magazynowych,
* Zapisywania dat realizacji zamówień i ich statusów.

**Krok 1: Projektowanie nowych tabel w bazie danych**

**Zadanie 1:** Wywołaj setup.py, które utworzą nowe tabele, dodaj 10 nowych klientów według schematu przedstawionego w skrypcie. Otwórz generate\_operations.py i uruchom skrypt. Używając display.py sprawdź stan magazynu.

**Krok 2: Modyfikacja GUI**

**Zadanie 2:** W pliku app\_gui.py dodaj nową zakładkę: **"Zamówienia"**.

Powinna zawierać:

* Listę rozwijaną z klientami (Combobox).
* Listę dostępnych produktów z możliwością wyboru ilości (np. tabela lub pola tekstowe).
* Przycisk **„Złóż zamówienie”**.

W momencie kliknięcia przycisku:

* Program sprawdza, czy w magazynie jest wystarczająca ilość każdego produktu.
* Jeśli tak: zapisuje zamówienie, odejmuje ilość z magazynu i zapisuje operację sprzedaży z aktualną datą.
* Jeśli nie: wyświetla komunikat błędu o niedostępności.

**Krok 3: Logika realizacji zamówienia**

**Zadanie 3:** W module GUI (lub osobnym module) napisz funkcję realizującą zamówienie:

Funkcja powinna:

* Przyjąć ID klienta oraz listę produktów z ilościami (tzw. koszyk).
* Sprawdzić dostępność każdego produktu.
* Dodać rekord do Zamówienia i PozycjeZamówienia.
* Odjąć produkty z tabeli Produkty.
* Dodać operację magazynową (Sprzedaż, data, ilość).

Na koniec baza powinna zawierać:

* Nowe zamówienie,
* Zmieniony stan magazynowy,
* Nową operację magazynową (z typem Sprzedaż).

**Krok 4: Weryfikacja danych**

**Zadanie 4:**

* Wyświetl dane z tabeli Zamówienia i PozycjeZamówienia w zakładce GUI.
* Dodaj filtr np. po kliencie, dacie lub statusie.
* Zweryfikuj, że po złożeniu zamówienia stan magazynowy został pomniejszony.

**Podpowiedzi**

* Użyj datetime.now().isoformat() do zapisu daty.
* Operacje INSERT i UPDATE wykonuj w jednej transakcji (conn.commit() na końcu).
* Zadbaj o walidację danych wejściowych (czy klient został wybrany, ilość towaru to liczba > 0).

**Rezultat**

Po wykonaniu instrukcji system będzie umożliwiał:

* Obsługę klientów i zamówień,
* Sprawdzanie dostępności towarów,
* Rejestrowanie sprzedaży jako operacji magazynowej,
* Śledzenie dat i statusów realizacji.