

**О.В. Марковець, В.Г. Марковець\***

Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж,

\*Львівська філія дочірньої компанії “Газ України” НАК “Нафтогаз України”

## **АЛГОРИТМ ФОРМУВАННЯ ІНФОРМАЦІЇ В БАЗІ ДАНИХ РОЗРАХУНКІВ ПОКУПЦІВ З ПРОДАВЦЕМ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ**

© Марковець О.В., Марковець В.Г., 2008

**Запропоновано концептуальну схему бази даних і алгоритм формування даних про стан розрахунків між Покупцем і Продавцем у режимі реального часу. Описано основні таблиці бази даних, подано визначення вхідних та похідних даних системи розрахунків. Наведено блок-схеми і опис алгоритмів формування похідних даних.**

**In the article a conceptual database scheme and data forming real-time algorithm of calculations between Buyer and Seller are given. Main databases tables are described bases given are input and output data of calculating system are described. Block-schemes and description of algorithms of output data forming are given.**

### **Постановка проблеми у загальному вигляді**

В існуючих базах даних інформація про стан розрахунків Покупців з Продавцем (залишки неоплачених товарів, залишки кредитів, оплачені товари, оплата в кредит та ін.) генерується зовнішніми програмними процедурами. Це вимагає ініціації цих процедур користувачем програмного забезпечення. Це означає, що ми не можемо гарантувати актуальність бази даних про стан розрахунків в довільний момент роботи системи розрахунків. Крім того, програмні процедури розрахунку даних знаходяться поза базою даних, що не гарантує правильність даних для різних прикладних систем, які використовують цю базу даних. Водночас сучасні СУБД мають засоби реагування на внесення, зміни і знищення даних користувачем. Це тригери вставки, зміни і знищення в таблицях бази даних. До бази даних вводяться користувачем дані (наприклад, кількість і ціна проданих товарів, суми оплачених коштів), тригери вставки відповідних таблиць ініціюють процедури розрахунку похідних даних (залишки, оплачені товари та ін.). Отже, використовуючи тригери таблиць бази даних, можна забезпечити розрахунок і запис до бази даних інформації про стан розрахунків в момент введення даних. А це забезпечує правильність і актуальність бази даних у довільний момент часу.

### **Опис системи розрахунків**

Розрахунки Покупців з Продавцем обліковують за операційними періодами (ОП). ОП – це період часу, що характеризується типом (день, тиждень, декада, квартал, півріччя, рік) і датою ОП. Тип ОП задається при створенні в базі даних нового Продавця і не може бути змінений, якщо до бази даних уже було внесено інформацію про розрахунки Покупця. Дата ОП – це календарний день закінчення операційного періоду. При переході на наступний ОП до дати ОП додається кількість днів, що відповідають типу ОП. Дані в базу можуть вноситись і змінюватись тільки в поточному ОП Продавця. Для Продавця можна задавати декілька наборів угод з Покупцями, за якими проводитимуть облік розрахунків.

Дані, що відображають стан розрахунків Покупця з Продавцем, є двох видів:

- вхідні дані;
- похідні дані.

Вхідні дані вносяться користувачем, а похідні розраховуються системою.

До вхідних даних належать:

- продажі (Кпр);
- оплати (Сопл);
- повернені продажі (Кппр);
- повернені кошти (Спк).

До похідних даних належать:

- оплачені товари з кредиту (Козкр);
- оплачені товари (Кот) ;
- оплата в кредит (Сокр);
- повернені товари (Кпт);
- повернені неоплачені товари в ОП повернення (Кпнт);
- повернені неоплачені товари в ОП продажу;
- повернені оплачені товари з кредиту (Кпоткр);
- повернені оплачені товари (Кпот);
- повернені оплати (Спо);
- повернені оплати з кредиту в ОП повернення (Спокр);
- повернені оплати з кредиту в ОП оплати;
- повернені оплати з оплачених товарів (Споот);
- повернені оплати з оплачених товарів з кредиту (Спооткр);
- залишки неоплачених товарів на кінець ОП (Кнт);
- залишки кредитів на кінець ОП (Скр).

Вхідні та похідні дані в поточному ОП Продавця пов'язані залежностями:

$$\begin{aligned} \text{Кпр} &= \text{Козкр} + \text{Кот} + \text{Кнт} \\ \text{Сопл} &= \text{Кот} * \text{Цт} + \text{Сокр} \\ \text{Кппр} &= \sum \text{Кпт} \\ \text{Кпт} &= \text{Кант} + \text{Кпоткр} + \text{Кпот} \\ \text{Спк} &= \sum \text{Спо} \\ \text{Спо} &= \text{Спокр} + \text{Споот} + \text{Спооткр} \\ \text{Кнт} &= \text{Кпр} - \text{Кпнт} - \text{Кот} + \text{Споот} / \text{Цт} \\ \text{Скр} &= \text{Сокр} + \text{Кпот} * \text{Цт} - \text{Спокр}, \end{aligned}$$

де Цт – ціна товару.

### Структура бази даних

Спрощена структура бази даних розрахунків Покупця з Продавцем представлена на рис. 1.

На рисунку представлено таблиці, що входять до бази розрахунків Покупця з Продавцем і зберігають інформацію про введені дані користувачем та розраховані системою дані. Ця інформація повністю визначає стан розрахунків Покупця з Продавцем. В структурі таблиць представлені тільки ті поля, які використовуються для опису алгоритму. Розглянемо детальніше таблиці бази даних і зв'язки між ними.

Таблиці „Продавці”, „Покупці” описують дані Продавця і Покупця відповідно. До таблиці „Набори угод Продавця” вносяться коди і назви наборів угод, за якими проводяться розрахунки Покупців з Продавцем.

Таблиці „Продажі”, „Оплати”, „Повернені продажі”, „Повернені кошти” призначені для введення даних користувачем.

Таблиці „Оплачені товари з кредиту”, „Оплачені товари”, „Оплата в кредит”, „Повернені товари”, „Повернені неоплачені товари в ОП повернення”, „Повернені неоплачені товари в ОП продажу”, „Повернені оплачені товари з кредиту”, „Повернені оплачені товари”, „Повернені оплати”, „Повернені оплати з кредиту в ОП повернення”, „Повернені оплати з кредиту в ОП оплати”, „Повернені оплати з оплачених товарів”, „Повернені оплати з оплачених товарів з

кредиту”, „Залишки неоплачених товарів на кінець ОП”, „Залишки кредитів на кінець ОП” призначені для запису проміжних даних.

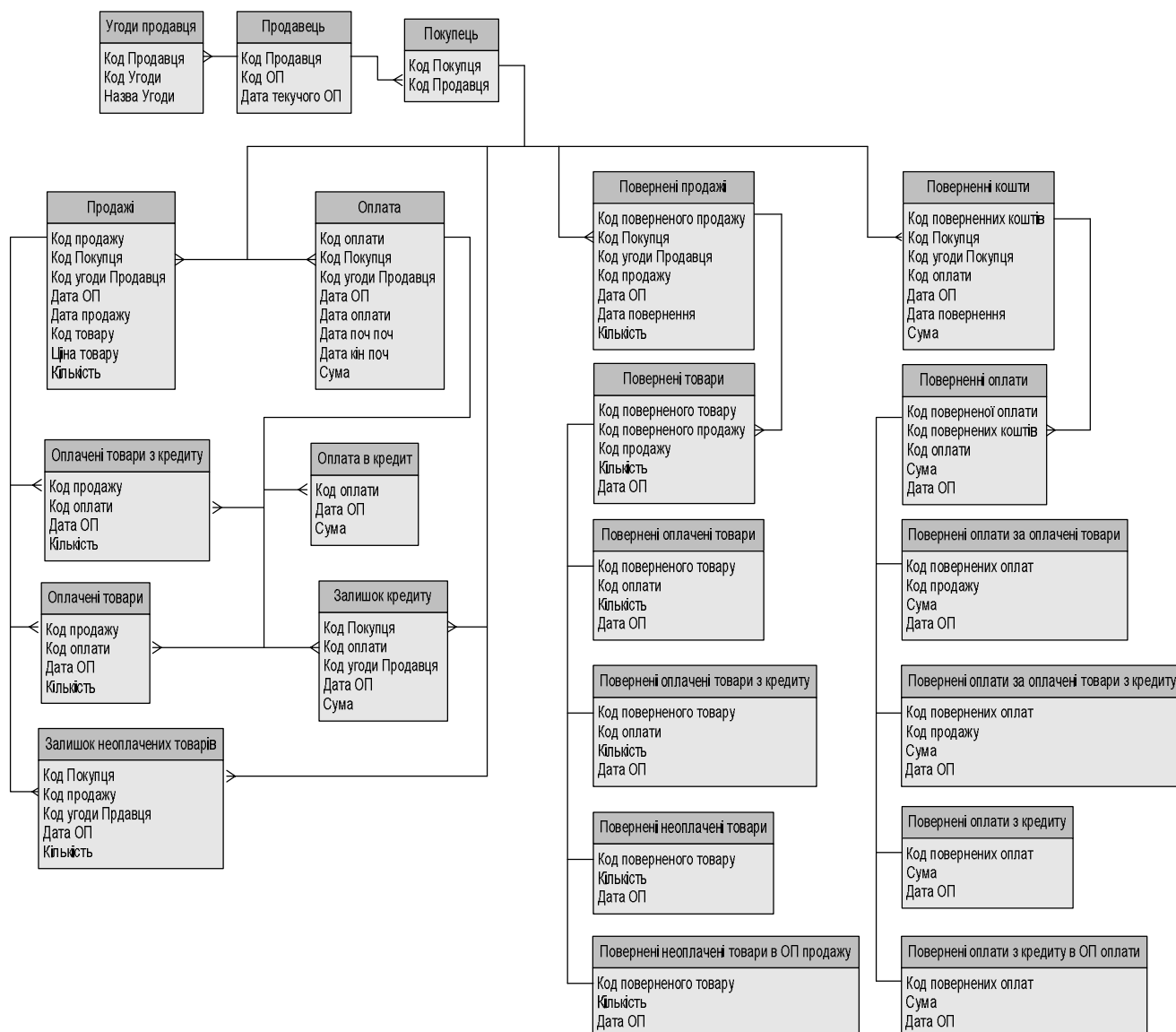


Рис. 1. Спрощена структура бази даних розрахунків Покупця з Продавцем

Коди ключових полів таблиць „Продавці”, „Покупці”, „Продажі”, „Оплати”, „Повернені продажі”, „Поверненні кошти”, „Поверненні товари”, „Поверненні оплати” генеруються системою при внесенні нового запису в таблицю і на рівні поля захищені від будь-яких змін.

Тригери вставки, поновлення і знищення в таблицях „Продажі”, „Оплати”, „Повернені продажі”, „Поверненні кошти” формують і вносять похідні дані у відповідні таблиці. Тригери вставки, поновлення і знищення в таблиці „Продавці” забезпечують перехід на наступний операційний період і знищення Продавця, який не має жодного Покупця з оборотами у будь-якому ОП. Тригер знищення в таблиці „Покупці” знищує Покупця, який не має жодних оборотів у будь-якому ОП. Вставка і поновлення даних у вхідних і похідних таблицях проводиться через механізм транзакцій, що забезпечує цілісність бази даних та її мережеве використання.

Всі таблиці на рівні запису захищені від будь-яких змін, які проводяться не у поточному ОП Продавця. При цьому введений механізм блокування контролю для забезпечення реіндексації і зміни структури таблиць (за необхідності) у непустій базі даних.

## Загальний опис алгоритму

Опишемо детальніше вхідні і похідні дані.

**Вхідні дані.** **Продажі** вносяться на основі документа (реєстр реалізації, акт прийому-передачі, накладна та ін.) – код угоди Продавця, дата продажу, код товару, ціна, кількість; дата ОП, код продажу, код Продавця генеруються системою. **Оплати** вносяться на основі документа (банківська виписка, авізо, протокол та ін.) – код угоди Продавця, дата оплати, сума, дата початку і закінчення погашення боргу; дата ОП, код продажу, код Продавця генеруються системою. **Повернені продажі** можуть вноситись двома способами – задається тільки кількість повернених товарів, код набору угод Продавця і дата повернення, а система вибирає конкретні продажі для повернення з врахуванням попередніх повернень або користувач вказує конкретний продаж для повернення. Аналогічно **Повернені кошти** вносяться двома способами – внесення суми повернених коштів або вибір конкретної оплати для повернення.

**Похідні дані.** **Залишки неоплачених товарів (НТ)** на кінець ОП формуються з вхідних залишків НТ і внесення продаж. Протягом ОП залишки НТ змінюються при внесенні інших вхідних даних. **Залишки кредитів** на кінець ОП формуються з вхідних залишків на початок ОП і внесених оплат і змінюються при внесенні інших вхідних даних. **Оплачені товари з кредиту** формуються в ОП при внесенні продажів, якщо є вхідний кредит і визначають кількість (сума\_вх\_кр/ціна). **Оплачені товари** формуються в ОП при внесенні оплат і визначають кількість (сума\_опл/ціна) для залишків НТ на початок ОП і продаж в ОП. **Оплата в кредит** формується в ОП при внесенні оплат і змінюється в ОП при внесенні інших вхідних даних. **Повернені товари** формуються під час оброблення повернених продажів, **Повернені оплати** формуються під час оброблення повернених коштів. Похідні дані **Повернені неоплачені товари в ОП повернення, Повернені неоплачені товари в ОП продажу, Повернені оплачені товари з кредиту, Повернені оплачені товари, Повернені оплати з кредиту в ОП повернення, Повернені оплати з кредиту в ОП оплати, Повернені оплати з оплачених товарів з кредиту, Повернені оплати з оплачених товарів** формуються під час оброблення повернених товарів і повернених оплат відповідно, враховуючи стан розрахунків на момент повернення.

## Тригери таблиць

За результатами дії тригери поділяються на дві групи: тригери таблиць суб'єктів розрахунків і тригери таблиць вхідних даних. Алгоритм тригера таблиці „Продавці” наведено на рис. 2.

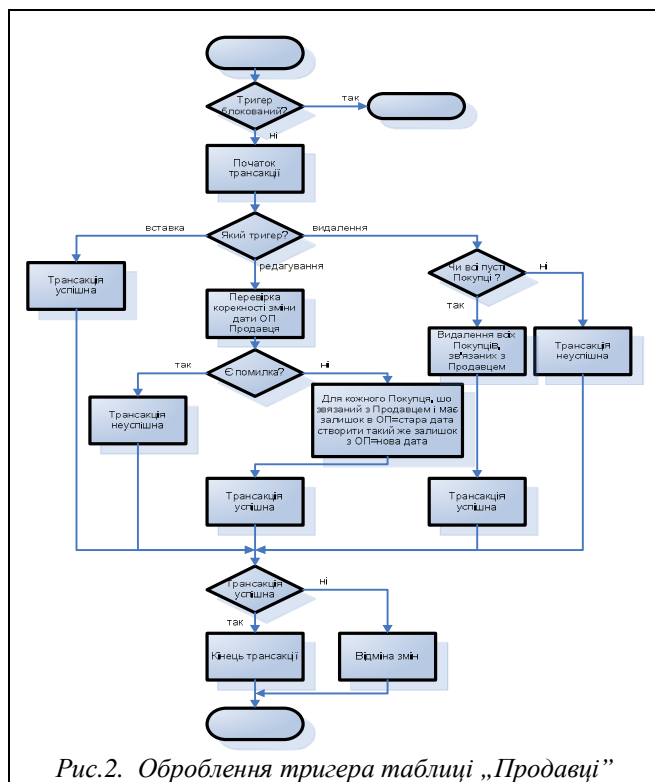


Рис.2. Оброблення тригера таблиці „Продавці”

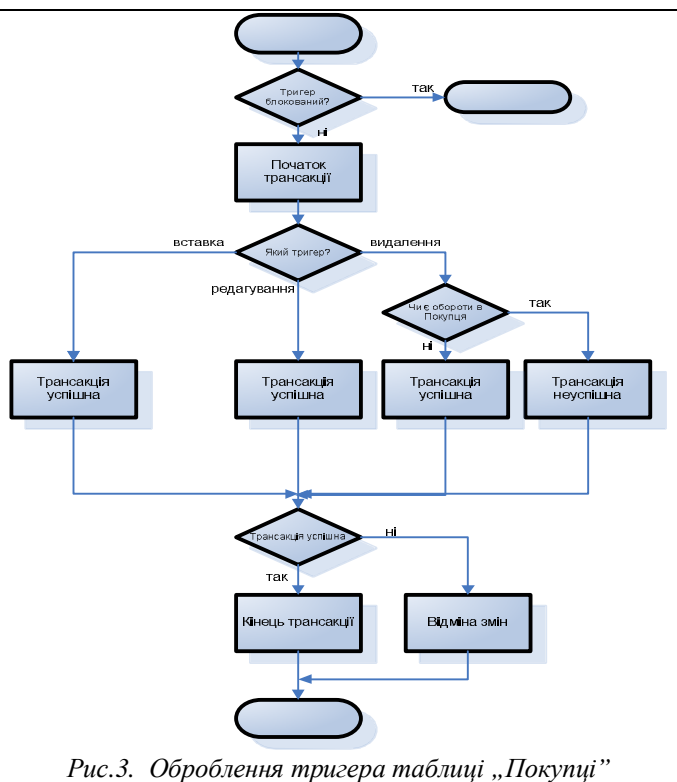


Рис.3. Оброблення тригера таблиці „Покупці”

Після перевірки правильності зміни дати ОП запускається процедура перенесення залишків НТ і залишків кредитів всіх Покупців, що пов'язані з Продавцем з поточного ОП на новий. Видалення Продавця проводиться після видалення всіх Покупців, що пов'язані з Продавцем за умови, що жоден з Покупців не має оборотів за будь-який ОП. Алгоритм тригера таблиці „Покупці” представлений на рис.3. Тригер видаляє Покупця, якщо він не має жодних оборотів за будь-який ОП. Дії з формування і запису похідних даних у відповідні таблиці ініціюються тригерами таблиць, які зберігають вхідні дані. Алгоритми тригерів наведено на рис.4–7.

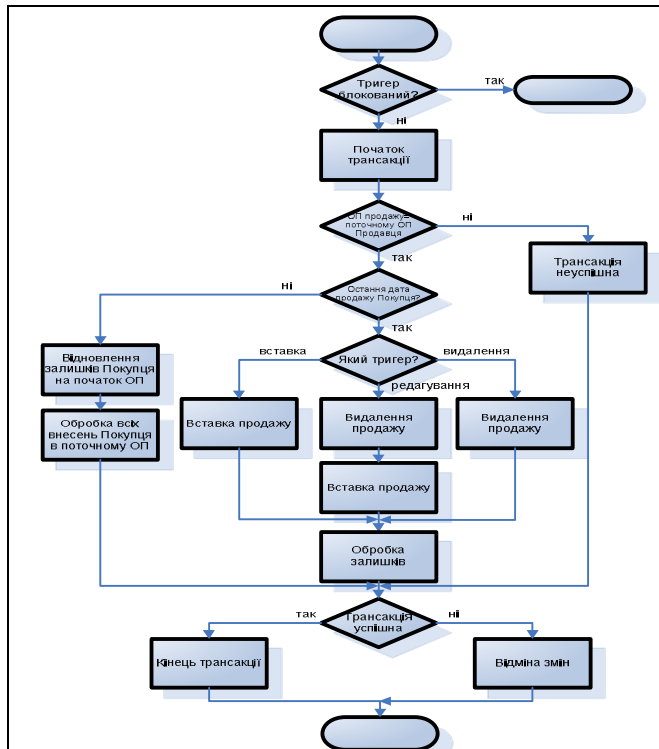


Рис. 4. Оброблення тригера таблиці „Продажі”

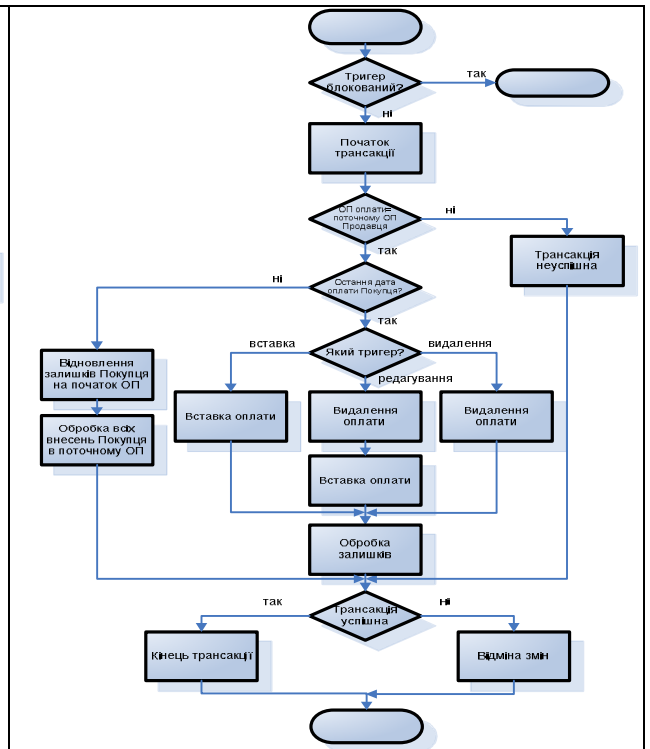


Рис. 5. Оброблення тригера таблиці „Оплати”

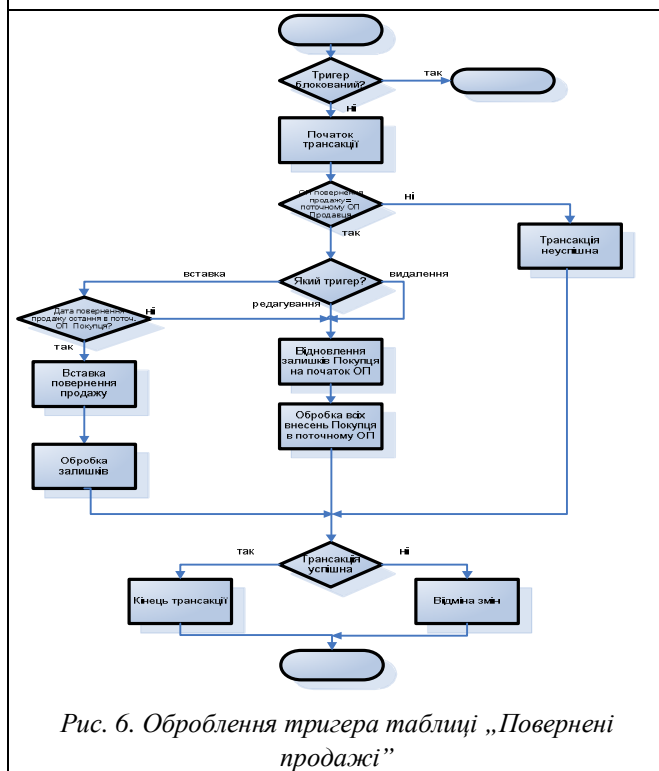


Рис. 6. Оброблення тригера таблиці „Повернені продажі”

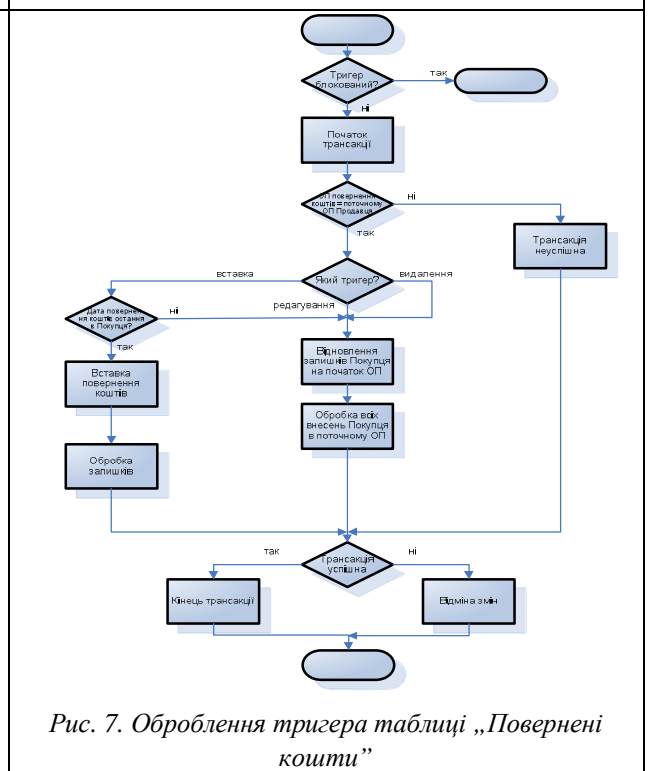


Рис. 7. Оброблення тригера таблиці „Повернені кошти”

У цих тригерах застосовано два методи обробки:

- „відкат” + оброблення всіх внесень для Покупця в поточному ОП і оброблення залишків;
- оброблення конкретного (вставка, зміна, видалення) внесення Покупця в поточному ОП і оброблення залишків.

„Відкат” – це дія системи з відновлення залишків Покупця в поточному ОП за залишками попереднього ОП і обнулення похідних даних поточного ОП. Внесення – це дія користувача із вставляння, зміни або видалення продажу, оплати, повернення продажів або повернення коштів. Під вставлянням розуміють як нове внесення, так і відновлення після видалення.

Умова вибору методу оброблення – дата внесення. Для тригерів „Продажі” і „Оплати” перший метод використовується тільки за умови, що дата внесення не найбільша для Покупця у поточному ОП. Для тригерів „Повернення продаж” і „Повернення коштів” другий метод застосовується тільки для вставляння і за умови, що дата внесення найбільша для Покупця в поточному ОП.

Оброблення залишків – це порівняння залишків кредитів і залишків неоплачених товарів Покупця в поточному ОП і визначення похідних даних – „оплачені товари з кредиту”, „оплачені товари”, „оплачено в кредит” з відповідним корегуванням залишків.

### **Опис процедур, що забезпечують роботу тригерів таблиць вхідних даних**

Для забезпечення роботи тригерів застосовано такі процедури:

- відновлення залишків Покупця на початок ОП;
- оброблення всіх внесень Покупця у поточному ОП;
- вставка продажу;
- видалення продажу;
- вставка оплати;
- видалення оплати;
- вставка повернення продажу;
- вставка повернення коштів;
- оброблення залишків.

Процедура „Відновлення залишків Покупця на початок ОП” складається з трьох етапів:

- видалення залишків (кредитів і неоплачених товарів) Покупця в поточному ОП;
- відновлення залишків Покупця в поточному ОП за залишками попереднього ОП;
- видалення даних для поточного ОП у таблицях „Оплачені товари з кредиту”, „Оплачені товари”, „Оплачено в кредит”, „Повернені товари”, „Повернені оплати”, „Повернені неоплачені товари в ОП продажу”, „Повернені неоплачені товари в ОП повернення”, „Повернені оплачені товари з кредиту”, „Повернені оплачені товари”, „Повернені оплати з кредиту в ОП оплати”, „Повернені оплати з кредиту в ОП повернення”, „Повернені оплати з оплачених товарів з кредиту”, „Повернені оплати з оплачених товарів”.

Процедурою „Оброблення всіх внесень Покупця в поточному ОП” переглядають календарні дні поточного ОП, починаючи з першого і для кожного дня виконуються процедури вставки у послідовності – „вставляння продажу”, „оплати”, „повернення продажу”, „повернення коштів”, якщо дата вставляння дорівнює поточному дню обробки. Перед процедурами „вставляння повернення продажу” і „вставляння повернення коштів” виконується процедура „оброблення залишків”. Після перегляду всіх днів ОП виконується процедура „оброблення залишків”. Алгоритм процедури „оброблення всіх внесень Покупця в поточному ОП” представлена на рис. 8.

Процедура „вставляння продажу” додає до таблиці „Залишки неоплачених товарів” кількість товарів з кодом продажу.

Процедура „видалення продажу” видаляє з таблиць „Залишки неоплачених товарів”, „Оплачені товари з кредиту” і „Оплачені товари” записи з кодом продажу і додає в таблицях „Залишки кредитів” і „Оплачено в кредит” суми за вартістю видалених товарів з кредиту і оплачених товарів.

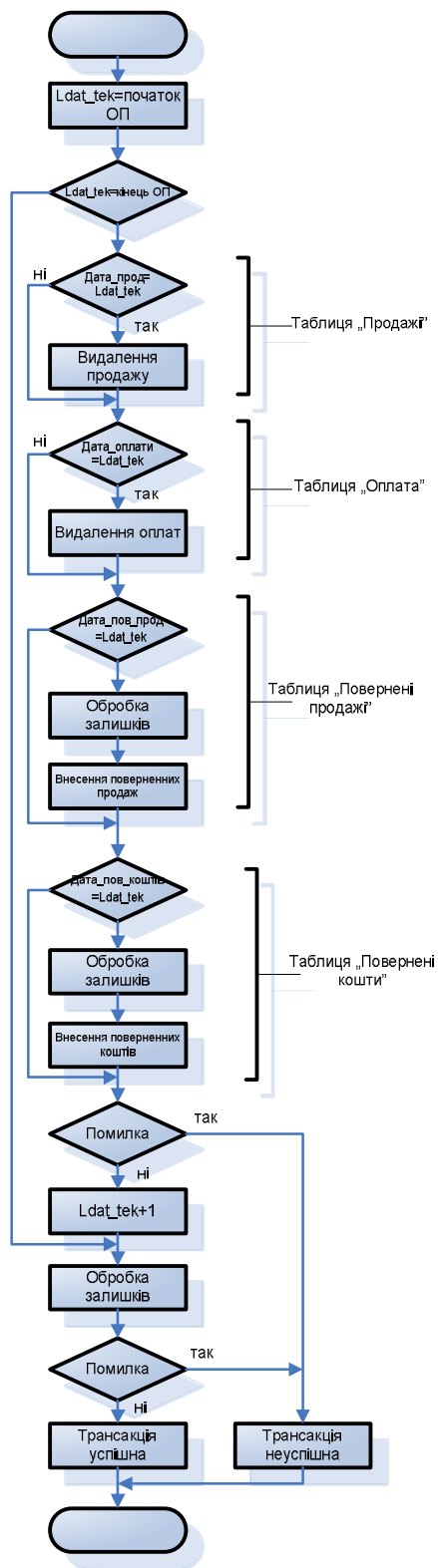


Рис. 8. Оброблення всіх внесень Покупця в поточному ОП

Процедура „вставляння оплати” додає в таблиці „Залишки кредитів” і „Оплачено в кредит” суму оплати з кодом оплати.

Процедура „видалення оплати” видаляє з таблиць „Залишки кредитів”, „Оплачено в кредит”, „Оплачені товари” записи з кодом оплати і додає до таблиці „Залишки неоплачені товари” кількість видалених оплачених товарів.

Алгоритм обробки процедури „вставлення повернення продажів” наведено на рис. 9.

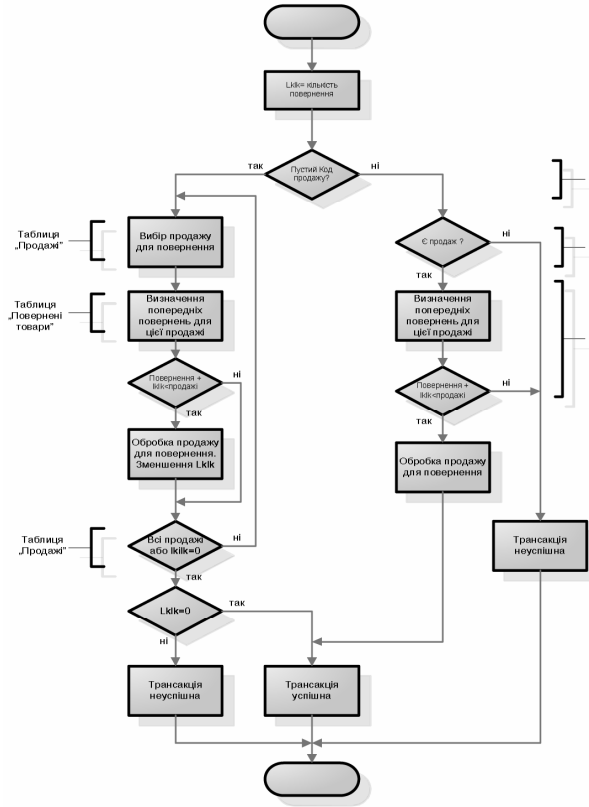


Рис. 9. Вставлення повернення продаж

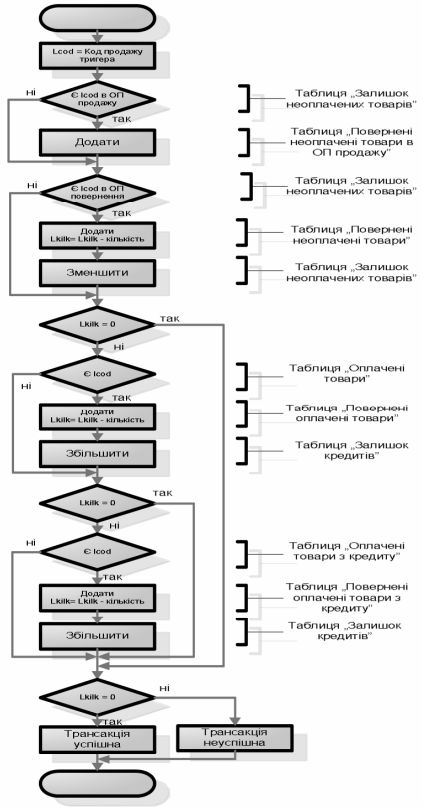


Рис. 10. Оброблення продажу для повернення

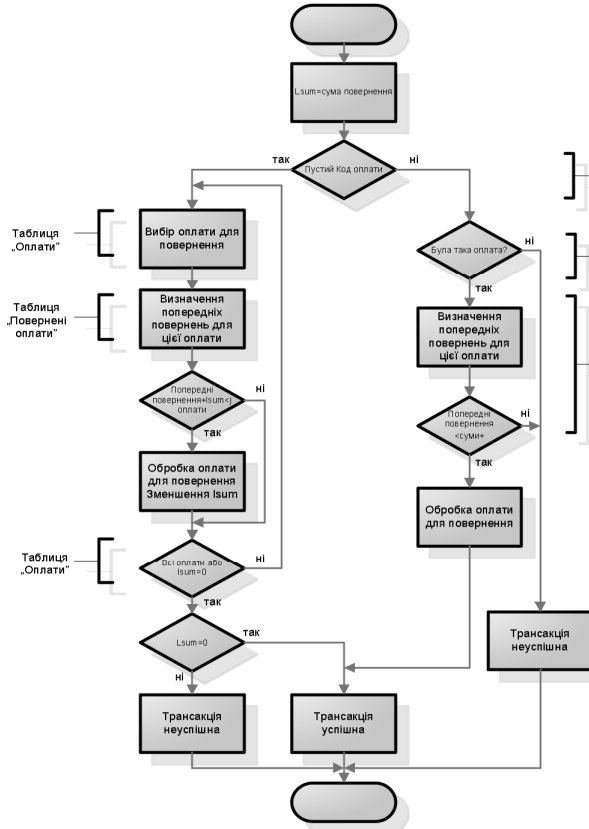


Рис. 11. Вставлення повернення коштів

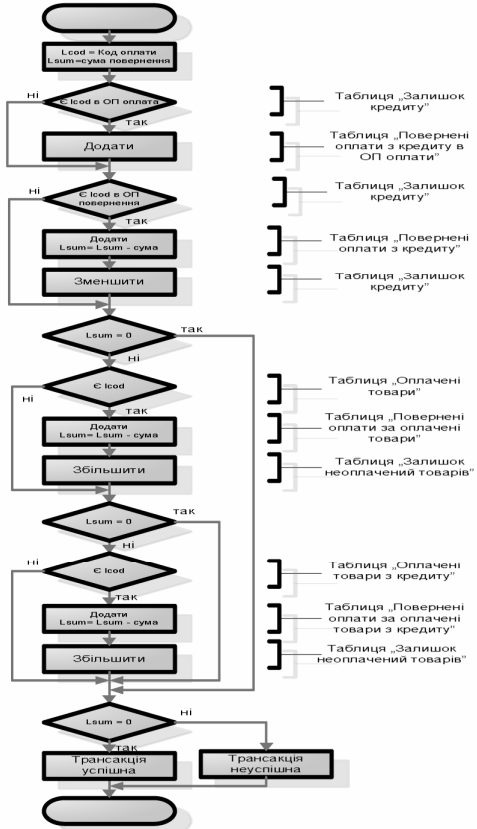


Рис. 12. Оброблення оплати для



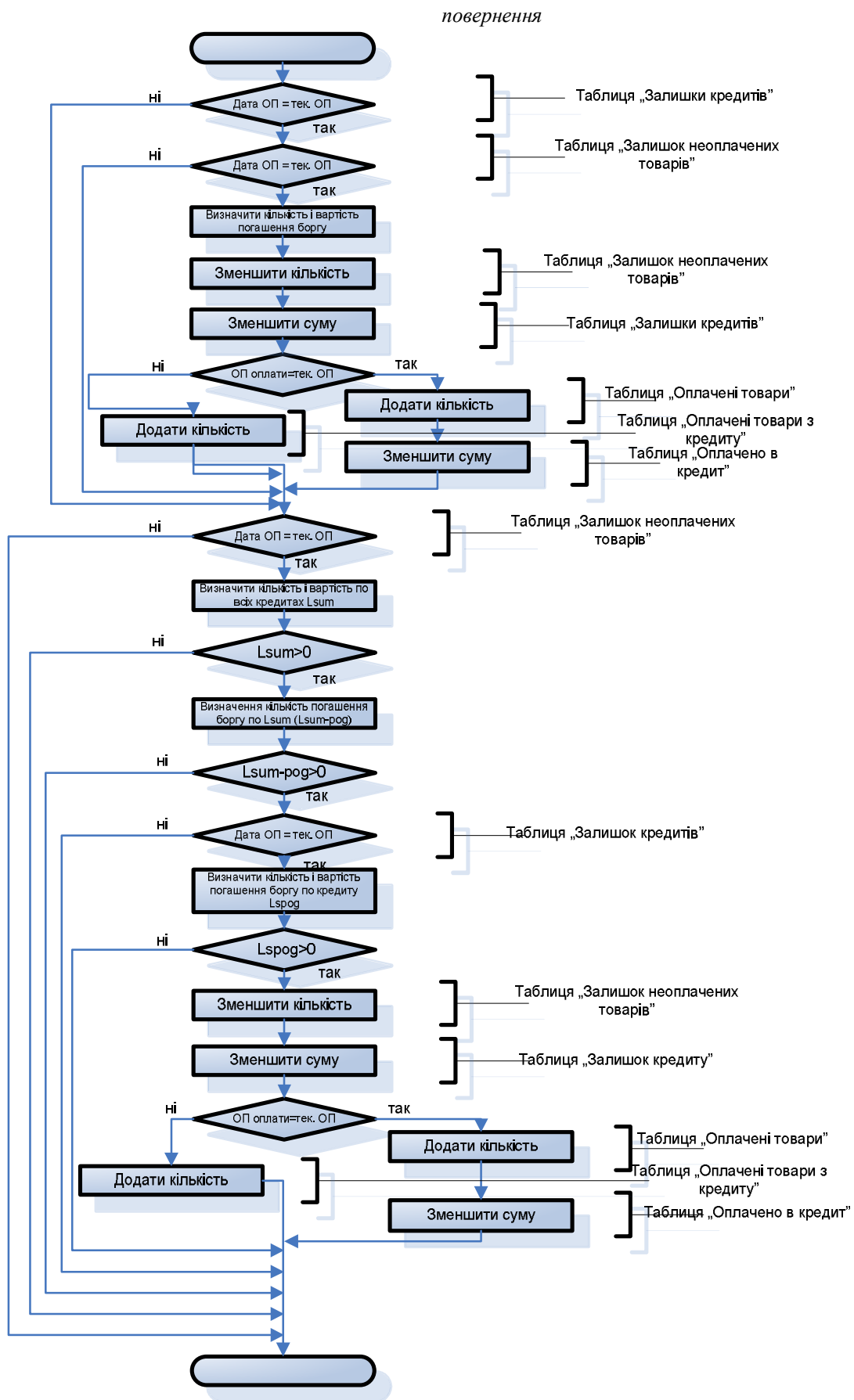


Рис. 13. Оброблення залишків Покупця

Процедура „вставлення повернення продажів” використовує процедуру „оброблення продажу для повернення”, алгоритм якої представлений на рис. 10.

Алгоритми оброблення процедури „вставлення повернення коштів” і процедури „оброблення оплати для повернення” представлені на рис. 11–12.

Алгоритм оброблення процедури „оброблення залишків” представлений на рис.13.

У процесі оброблення залишків забезпечуються можливості погашення заборгованості за параметрами, які вказуються при внесенні оплат:

- погашення боргу у межах дат реалізації товарів (початок і кінець погашення боргу);
- якщо для Покупця встановлена можливість погашення боргу поза межами початку і кінця погашення боргу, забезпечується можливість погашення таких боргів;
- заборгованість поштучних товарів може погашатись деякою кількістю оплат.

Процес оброблення проходить у два етапи:

– на першому етапі переглядаються залишки кредитів, для кожного кредиту переглядаються всі залишки неоплачених товарів і погашаються борги за визначеними параметрами при внесенні оплати;

– на другому етапі переглядаються залишки неоплачених товарів, для кожного залишку визначається загальна сума кредитів Покупця і погашається борг, якщо виконуються умови погашення боргу.

### **Висновки**

Отже, використовуючи той факт, що Visual FoxPro – це об’єктно-орієнтована мова, яка керується за подіями, можна створити базу даних розрахунків, що працює в режимі реального часу і забезпечує повну інформацію про стан розрахунків Покупця з Продавцем

**УДК 51.001.57+004.652.4+004.827**

**В.В. Пасічник, Н.Б. Шаховська**  
Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## **ІНФОРМАЦІЙНІ КОМПОНЕНТИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ**

© Пасічник В.В., Шаховська Н.Б., 2008

**Показано відмінності між базами, сховищами та просторами даних. Введено означення інженерії знань та інтелектуальної системи.**

**Differences are shown between databases, datawarehouses and dataspaces. D of engineering of knowledges and intellectual system is entered.**

### **Вступ**

Інтелектуальні системи є одними із найпоширеніших засобів підтримки прийняття рішень. Проте сьогодні немає чіткого визначення цього терміна, опису складових таких систем та їхніх завдань. Тому подамо ряд визначень, які характеризують інтелектуальні системи, та охарактеризуємо їх складові.