

Wyzwalacze, funkcje serwerowe, widoki

1. Zapisz **funkcję wyzwalacza** (trigger) **kontr_stan** w **swojej** bazie rembud:

-- baza rembud, funkcja trigger - kontrola zerowych stanów magazynu

```
declare
oldstan numeric(8,2);
nazproduktu text;

begin
select into oldstan stan from produkty where idproduktu = NEW.idproduktu;
select into nazproduktu nazwa from produkty where idproduktu = NEW.idproduktu;
if NEW.ilosc > oldstan then
    raise exception 'zabraklo towaru: %',nazproduktu;
end if;
return NEW;
end;
```

Skonstruuj **wyzwalacz wkontr_stan** nie dopuszczający do sprzedaży towarów, których nie ma w magazynie w wystarczającej ilości

Sprawdź działanie wyzwalacza za pomocą odpowiedniego zapytania SQL.

2. Napisz funkcję wyzwalacza **mod_mag** oraz wyzwalacz **wmod_mag** aktualizujący stan magazynu o sprzedane ilości produktów w reakcji na wprowadzanie informacji o sprzedaży.

Sprawdź działanie wyzwalacza.

3. Zapisz **funkcję serwerową klient_zaplacil(id_klienta, data_pocz, data_konc)** wyliczającą ile łącznie dany klient zapłacił w sklepie w okresie od data_pocz do data_konc:

```
begin
SELECT INTO zaplacil SUM(ilosc*cena*(1+vat))
FROM nagsprzedaz NATURAL JOIN pozsprzedaz
NATURAL JOIN produkty
WHERE idklienta=id_klienta AND datasp >= data_pocz AND datasp <= data_konc;
End;
```

Wypróbuj działanie funkcji w zapytaniach:

- SELECT klient_zaplacil('K01', '2020-03-01', '2021-03-31');
- SELECT id klienta, klient_zaplacil(id klienta, '2021-03-15', '2021-03-31') FROM klienci;
- Wykorzystaj funkcję **klient_zaplacil** w zapytaniu: Podaj wartości styczniowych zakupów poszczególnych klientów pochodzących z Gdyni

4. Zdefiniuj **funkcję serwerową licz_brutto** obliczającą wartość brutto dla **podanej wartości netto i vat**. Sformułuj zapytanie wyliczające wartości brutto poszczególnych produktów i wykorzystujące zdefiniowaną funkcję

5. Zdefiniuj **widok rejfakt** tak aby wirtualna tabela zawierała następujące dane **o fakturach**: {nazwa klienta, nrfaktury, data sprzedaży, wartość netto, wartość vat, wartość brutto}

Wykorzystaj widok w zapytaniach:

- Podaj wartość podatku VAT jaki zapłacili poszczególni klienci (nazwa klienta, podatek)
- Podaj wartości netto i brutto zarobione przez rembud w poszczególnych miesiącach.

6. Zmodyfikuj **funkcję kontr_stan** z punktu 1 tak, aby odrzucona sprzedaż została zapisana w tabeli **zamow** (idzamow, datazamow, idproduktu, ilosc), gdzie **idzamow** jest kolejnym numerem, **datazamow** bieżącą datą, a **idproduktu** i **ilosc** podanymi w instrukcji **INSERT** identyfikatorem i ilością produktu.

7. Skonstruuj **wyzwalacz**, który nie dopuści do sprzedaży czegokolwiek klientowi, który jest zadłużony w rembud na więcej niż 500 zł (łączna wartość brutto jego niezapłaconych faktur wynosi co najmniej 500 zł)

8. Napisz **wyzwalacz**, który uniemożliwi obniżenie ceny produktu o więcej niż 10%

9. Zdefiniuj widok **prodetal**(idproduktu, nazproduktu, cena_brutto, stan). Wykorzystaj widok do:

- policzenia należności klienta K03 za zakupy zrealizowane w pierwszej połowie stycznia
- wskazania nazwy produktu z najwyższą ceną brutto

10. Zdefiniuj **funkcję serwerową pierwszy_kontakt**(id_klienta), która dla podanego identyfikatora klienta poda datę wystawienia jego pierwszej (najstarszej) faktury. Wykorzystaj ją w zapytaniach:

- Kiedy wystawiono pierwszą fakturę klientowi K05?
- Podaj, kiedy po raz pierwszy wystawiono faktury poszczególnym klientom z Sopotu
- Podaj nazwy klientów, którzy po raz pierwszy odwiedzili sklep później niż klient K03

11. Zdefiniuj **funkcję serwerową srednia_producenta**(nazproducenta) wyliczającą średnią cenę produktów dostarczanych przez danego producenta. Wykorzystaj ją w zapytaniach:

- podaj średnią cenę produktów Cersanita
- podaj średnią cenę produktów u producentów dostarczających wiertarki (nazwa producenta, średnia cena jego produktów)

12. Zdefiniuj **funkcję serwerową wart_faktury**(numfaktury), która dla podanego numeru faktury wylicza wartość brutto tej faktury. Wypróbuj działanie funkcji w zapytaniach:

- Policz średnią wartość faktury wystawianej w sklepie rembud.
- Podaj nazwy klientów (i numery ich faktur), których jednorazowe zakupy warte były więcej niż średnia wartość faktury.

Spróbuj powyższe zapytania zapisać w SQL bez użycia funkcji wart_faktury

13. Zdefiniuj funkcję **wart_producent(dzien, nazproducenta)**, która dla zadanej daty i nazwy producenta poda wartość sprzedaży produktów danego producenta w danym dniu. Jeśli w danym dniu nie sprzedano żadnego produktu wskazanego producenta funkcja powinna zwrócić wartość 0. Wypróbuj działanie funkcji w zapytaniu:

- Podaj daty dni, kiedy producent Cersanit zarobił więcej niż 500 (dopasuj wartość).
- Sformułuj powyższe zapytanie bez używania funkcji wart_producent

14. Zdefiniuj wyzwalacz, **dodaj_klienta** który podczas wprowadzania nagłówka sprzedaży dla klienta niezapisanego w tabeli klienci, najpierw zapisuje jego identyfikator w tabeli klienci, a następnie wprowadza nagłówek sprzedaży. (być może będzie to wymagało usunięcia więzów integralnościowych między odpowiednimi tabelami)

15. Zdefiniuj wyzwalacz, **poz10** który nie pozwala zapisać dla jednego dokumentu sprzedaży więcej niż 10 pozycji. Przy próbie zapisu kolejnej pozycji tworzony jest nowy nagłówek i kolejne pozycje są wiązane z nowym nagłówkiem (ewentualnie z jakimś komunikatem informującym) (dla celów testowych można założyć mniejszą niż 10 liczbę pozycji)

16. Zdefiniuj wyzwalacz **zmklient**, który przy jakiegokolwiek zmianie w tabeli klienci zapisuje starą wersję tego wiersza w tabeli archklienci, zapisując dodatkowo datę dokonania zmiany.

17. Napisz wyzwalacz, który będzie pilnował, aby liczba produktów Black&Deckera i Boscha razem wziętych nie przekraczała liczby produktów Stanleya

18. Zdefiniuj tabelę produkty1 o strukturze takiej, jak tabela produkty z dodatkową kolumną stan_minimalny (typ danych jak stan). Kolumnę stan_minimalny wypełnij wartościami równymi 5% stanu bieżącego. Następnie skonstruuj wyzwalacz, który w sytuacji sprzedaży zakończonej zejściem poniżej stanu minimalnego wyprowadzi stosowny komunikat.