Funkcje PL/SQL użyte w bazie danych

1.Procedura dodająca liczbę egzemplarzy danej książki

create or replace

procedure dodaj\_do\_stanu(

idk in ksiazki.idksiazki%type,

dodajilosc in ksiazki.stan%type) is

begin

update ksiazki set stan = stan+dodajilosc where idksiazki=idk;

end;

2.Procedura, która pozwala na dodanie przez użytkownika recenzji do danej książki

create or replace

procedure dodaj\_recenzje(

R\_TEKST IN RECENZJE.TEKST%TYPE,

R\_IDKLIENTA IN RECENZJE.IDKLIENTA%TYPE,

R\_IDKSIAZKI IN RECENZJE.IDKSIAZKI%TYPE) IS

R\_DATADODANIA DATE;

BEGIN

select SYSDATE into R\_DATADODANIA from dual;

INSERT INTO RECENZJE VALUES(RECENZJE\_SEQID.NEXTVAL,R\_TEKST,R\_DATADODANIA,R\_IDKLIENTA,R\_IDKSIAZKI);

END;

3. Procedura pozwalająca użytkownikowi na zmianę obecnego hasła na nowe

create or replace

procedure nowehaslo(

idk IN KLIENCI.IDKLIENTA%TYPE,

nowe\_haslo IN KLIENCI.HASLO%TYPE) IS

BEGIN

UPDATE KLIENCI SET HASLO= nowe\_haslo where IDKLIENTA = idk;

END;

4. Procedura pozwalająca na utworzenie nowego użytkownika

create or replace

PROCEDURE nowy\_klient(

K\_LOGIN IN KLIENCI.LOGIN%type,

K\_HASLO IN KLIENCI.HASLO%type,

K\_IMIE IN KLIENCI.IMIE%type,

K\_NAZWISKO IN KLIENCI.NAZWISKO%type,

K\_MIEJSCOWOSC IN KLIENCI.MIEJSCOWOSC%type,

K\_KOD IN KLIENCI.KOD%type,

K\_ULICA IN KLIENCI.ULICA%type,

K\_EMAIL IN KLIENCI.EMAIL%type) AS

BEGIN

INSERT INTO KLIENCI VALUES(KLIENCI\_SEQID.NEXTVAL, K\_LOGIN, K\_HASLO, K\_IMIE, K\_NAZWISKO, K\_MIEJSCOWOSC, K\_KOD, K\_ULICA, K\_EMAIL);

END nowy\_klient;

5. Procedura pozwalająca na usunięcie książki z koszyka

create or replace

procedure usun\_artykul(

idk in artykuly.idksiazki%type,

idzam in artykuly.idzamowienia%type,

ilosc in artykuly.idzamowienia%type) is

begin

delete from artykuly where idksiazki = idk and idzamowienia = idzam;

dodaj\_do\_stanu(idk,ilosc);

end;

6. Funkcja pozwalająca użytkownikowi na zmianę liczby egzemplarzy zamawianej książki znajdującej sie w koszyku klienta

create or replace

function zmien\_ilosc(

idz in artykuly.idzamowienia%type,

idk in artykuly.idksiazki%type,

nowailosc in artykuly.sztuk%type

aktualnailosc in artykuly.sztuk%type) is

ilosc in artykuly.sztuk%type;

begin

if aktualnailosc<nowailosc then

ilosc:=nowailosc-aktualnailosc;

if stan(idk,ilosc) then

update artykuly set sztuk = sztuk+ilosc where idzamowienia = idz and idksiazki = idk;

pomniejsz\_stan(idk,ilosc);

return 1;

end if

return -1;

end if;

if aktualnailosc>nowailosc then

ilosc:=aktualnailosc-nowailosc;

dodaj\_do\_stanu(idk,nowailosc);

return 1;

else

return 2;

end if;

end;

7. Funkcja, która wylicza całkowity koszt zamawianych książek znajdująch się w koszyku

create or replace

function cena\_zakupu(

idzam in artykuly.idzamowienia%type) return number is

cenazam number(7,2);

ii integer;

xx integer;

begin

select artykuly.idzamowienia, sum(ksiazki.cena\*artykuly.sztuk) into ii,cenazam from artykuly join ksiazki on artykuly.idksiazki = ksiazki.idksiazki

where idzamowienia=idzam group by artykuly.idzamowienia;

return cenazam;

end;

8. Funkcja sprawdzająca ilość egzemplarzy danej ksiązki na stanie magazynu i jej modyfikację

create or replace

function stan(

id\_książki IN KSIAZKI.IDKSIAZKI%TYPE,

liczba IN KSIAZKI.STAN%TYPE)

return boolean IS

ilość KSIAZKI.STAN%TYPE;

begin

select stan into ilość from ksiazki

where idksiazki = id\_książki;

if ilość >= liczba then

UPDATE ksiazki set stan=stan-liczba where idksiazki = id\_książki;

return true;

else

return false;

end if;

exception when no\_data\_found then

DBMS\_OUTPUT.Put\_line('Wystapił błąd, niepoprawne id');

end;

10. Funkcja wstawiająca dane do tabeli artykuły

create or replace

function wstaw\_artykuly(

id\_książki IN ksiazki.idksiazki%type,

liczba IN ksiazki.stan%type,

zam IN zamowienia.idzamowienia%type)

return integer IS

begin

if stan(id\_książki,liczba) then

insert into artykuly values (zam,id\_książki,liczba);

return 1;

else

return -1;

end if;

end;

11. Funkcja generująca id zamówienia. Wstawia dane do tabeli zamówienia.

create or replace

function zamow(

id\_klienta in zamowienia.idklienta%type) return integer is

x integer;

y integer;

data\_zamowienia date;

begin

select id\_zam.nextval into x from dual;

y := id\_zam.currval;

select SYSDATE into data\_zamowienia from dual;

insert into zamowienia values(y,data\_zamowienia,id\_klienta);

return y;

end;

12. Funkcja zwracająca id danego klienta

create or replace

function zwroc\_id(

k\_log in klienci.login%type) return integer is

k\_id klienci.idklienta%type;

begin

select idklienta into k\_id from klienci where login=k\_log;

return k\_id;

end;

13. Procedura zmniejszająca ilość książek w magazynie

create or replace

procedure pomniejsz\_stan(

idk in ksiazki.idksiazki%type,

ilosc in ksiazki.stan%type) is

begin

update ksiazki set stan = stan-ilosc where idksiazki=idk;

end;

12. Sekwencja generująca id zamówienia

create sequence id\_zam

minvalue 1

maxvalue 999999

start with 11

increment by 1

nocycle;

13.Sekwencja generująca id klienta

create sequence klienci\_seqid

minvalue 1

maxvalue 999999

start with 11

increment by 1

nocycle;

14. Sekwencja generująca id dodanej recenzji

create sequence recenzje\_seqid

minvalue 1

maxvalue 999999

start with 11

increment by 1

nocycle;