Aleksandra Osikowicz, Gabriela Osada, Kinga Kozik, Krzysztof Kubina

TABELE:

1. Klienci

**create** **or** replace **table** klienci(

id\_klient **INT** AUTO\_INCREMENT **primary** **key**,

imie **varchar**(50) **not** **null**,

nazwisko **varchar**(50) **not** **null**,

email **varchar**(100) **not** **null**,

data\_rejestracji **timestamp** **DEFAULT** NOW() **ON** **UPDATE** CURRENT\_TIMESTAMP

);

1. Status dostawy

**create** **or** replace **table** status\_dostawy(

id\_status **int** **primary** **key** auto\_increment,

status **varchar**(50)

);

1. Logowanie

**create** **or** replace **table** logowanie(

id\_logowanie **int** auto\_increment **primary** **key**,

login **varchar**(250) **unique** **not** **null**,

haslo **varchar**(30) **not** **null**,

id\_klient **int**,

**foreign** **key**(id\_klient) **references** klienci(id\_klient)

);

1. Książki info

**create** **or** replace **table** ksiazki\_info(

id\_ksiazki **int** auto\_increment **primary** **key**,

autor **varchar**(250),

tytul **varchar**(250),

rok\_wydania **int**,

isbn **varchar**(50),

epoka **varchar**(50)

);

1. Książki oceny

**create** **or** replace **table** ksiazki\_oceny(

id **int** auto\_increment **primary** **key**,

id\_ksiazki **int**,

ocena **float**(3,2),

liczba\_ocen **int**,

1\_gwiazdka **int**,

2\_gwiazdki **int**,

3\_gwiazdki **int**,

4\_gwiazdki **int**,

5\_gwiazdek **int**,

**foreign** **key**(id\_ksiazki) **references** ksiazki\_info(id\_ksiazki)

);

1. Zamówienia

**create** **or** replace **table** zamowienia(

id\_zamowienia **int** auto\_increment **primary** **key**,

id\_klient **int**,

data\_zamowienia **timestamp** **DEFAULT** NOW() **ON** **UPDATE** CURRENT\_TIMESTAMP,

id\_status **int**,

**foreign** **key**(id\_klient) **references** klienci(id\_klient),

**foreign** **key**(id\_status) **references** status\_dostawy(id\_status)

);

1. Ceny

**create** **or** replace **table** ceny(

id\_ksiazki **int**,

cena **float**(4,2) **primary** **key**,

**foreign** **key**(id\_ksiazki) **references** ksiazki\_info(id\_ksiazki)

);

1. Koszyk

**create** **or** replace **table** koszyk(

id\_zamowienia **int**,

id\_ksiazki **int**,

tytul **varchar**(250),

cena **float**(4,2),

**foreign** **key**(id\_zamowienia) **references** zamowienia(id\_zamowienia),

**foreign** **key**(id\_ksiazki) **references** ksiazki\_info(id\_ksiazki),

**foreign** **key**(cena) **references** ceny(cena)

);

1. Magazyn

**create** **or** replace **table** magazyn(

id\_magazyn **int** auto\_increment **primary** **key**,

id\_ksiazki **int**,

stan\_magazynowy **int**,

hurtownia **int**,

data\_aktualizacji **timestamp**,

**foreign** **key**(id\_ksiazki) **references** ksiazki\_info(id\_ksiazki)

);

1. Klienci kurier

**create** **or** replace **table** klienci\_kurier(

id\_kurier **INT** AUTO\_INCREMENT **primary** **key**,

id\_klient **int**,

id\_zamowienia **int**,

data\_zamowienia **timestamp** **DEFAULT** NOW() **ON** **UPDATE** CURRENT\_TIMESTAMP,

adres\_ulica **varchar**(50) **default** 'brak',

adres\_numer\_domu\_mieszkania **varchar**(10) **not** **null**,

adres\_miasto **varchar**(50) **not** **null**,

adres\_kod **varchar**(6) **not** **null**,

telefon **varchar**(9),

id\_status **int**,

**foreign** **key**(id\_klient) **references** klienci(id\_klient),

**foreign** **key**(id\_zamowienia) **references** zamowienia(id\_zamowienia),

**foreign** **key**(id\_status) **references** status\_dostawy(id\_status)

);

1. Epoki

**create** **or** replace **table** epoki(

id\_epoki **int**,

nazwa\_epoki **varchar**(100),

poczatek\_epoki **int** **primary** **key**,

koniec\_epoki **int**

)

**PARTITION** **BY** **RANGE** (poczatek\_epoki) (

**PARTITION** p\_przed\_renesansem **VALUES** LESS THAN (1493),

**PARTITION** p\_renesans **VALUES** LESS THAN (1521),

**PARTITION** p\_barok **VALUES** LESS THAN (1681),

**PARTITION** p\_oswiecenie **VALUES** LESS THAN (1789),

**PARTITION** p\_romantyzm **VALUES** LESS THAN (1851),

**PARTITION** p\_pozytywizm **VALUES** LESS THAN (1881),

**PARTITION** p\_modernizm **VALUES** LESS THAN (1919),

**PARTITION** p\_miedzywojenna **VALUES** LESS THAN (1939),

**PARTITION** p\_wspolczesnosc **VALUES** LESS THAN **MAXVALUE**

);

PODSTAWOWE SELECTY Z JOINAMI

1. Select wykonujący zliczanie ilości książek w każdym zamówieniu.

**SELECT**

zamowienia.id\_zamowienia,

**COUNT**(\*) **AS** *ile\_ksiazek*

**FROM** koszyk

**JOIN** zamowienia **ON** koszyk.id\_zamowienia = zamowienia.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**GROUP** **BY** zamowienia.id\_zamowienia;

1. Select, w którego rezultacie otrzymujemy listę 10 książek, posortowanych malejąco według oceny.

**SELECT**

tytul **AS** "tytuł",

ocena

**FROM** ksiazki\_info

**JOIN** ksiazki\_oceny **ON** ksiazki\_info.id\_ksiazki = ksiazki\_oceny.id\_ksiazki

**ORDER** **BY** ocena **DESC**

**LIMIT** 10;

1. Select sprawdzający stan magazynowy.

**SELECT**

tytul,

magazyn.stan\_magazynowy

**FROM** ksiazki\_info

**LEFT** **JOIN** magazyn **ON** magazyn.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** magazyn.hurtownia = 0

**ORDER** **BY** magazyn.stan\_magazynowy;

1. Select dający listę klientów i ich zamówionych książek.

**SELECT**

klienci.imie,

klienci.nazwisko,

ksiazki\_info.tytul

**FROM**

klienci

**JOIN**

zamowienia **ON** klienci.id\_klient = zamowienia.id\_klient

**JOIN**

koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN**

ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki;

1. Select wyświetlający zamówienia i adresy, pod które mają one być wysłane.

**SELECT**

zamowienia.id\_zamowienia,

klienci\_kurier.adres\_ulica,

klienci\_kurier.adres\_numer\_domu\_mieszkania,

klienci\_kurier.adres\_miasto,

klienci\_kurier.adres\_kod

**FROM**

zamowienia

**LEFT** **JOIN**

klienci\_kurier **ON** zamowienia.id\_klient = klienci\_kurier.id\_klient;

1. Select analizujący dostępność i ilość sprzedanych książek.

**SELECT**

ksiazki\_info.tytul,

**COUNT**(\*) **AS** *sprzedane*,

**CASE**

**WHEN** magazyn.hurtownia = 0 **THEN** 'brak w hurtowni'

**ELSE** 'dostępny w hurtowni'

**END** **AS** *dostepnosc*

**FROM** koszyk

**LEFT** **JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**LEFT** **JOIN** magazyn **ON** ksiazki\_info.id\_ksiazki = magazyn.id\_ksiazki

**GROUP** **BY** ksiazki\_info.id\_ksiazki

**ORDER** **BY** 2;

TRIGGERY

1. Trigger aktualizujący magazyn przed dodaniem pozycji do koszyka.

**create** **or** replace **trigger** aktualizacja\_magazynu **before** **insert** **on** koszyk

**for** **each** **row** **begin**

**update** magazyn

**set** stan\_magazynowy = stan\_magazynowy -1

**where** magazyn.id\_ksiazki = **new**.id\_ksiazki;

**end**;

1. Trigger tworzący login na podstawie imienia, nazwiska i id logowania.

**CREATE** **OR** REPLACE **TRIGGER** tworzenie\_loginu

**BEFORE** **INSERT** **ON** logowanie

**FOR** **EACH** **ROW**

**BEGIN**

**SET** **NEW**.login = CONCAT(

**LOWER**((**SELECT** imie **FROM** klienci **WHERE** id\_klient = **NEW**.id\_klient)),

'.',

**LOWER**((**SELECT** nazwisko **FROM** klienci **WHERE** id\_klient = **NEW**.id\_klient)),

**NEW**.id\_logowanie

);

**END**;

1. Trigger dodający epokę, w której napisana została książka przed dodaniem książki do księgarni.

**CREATE** **OR** REPLACE **TRIGGER** dodaj\_epoke

**BEFORE** **INSERT** **ON** ksiazki\_info

**FOR** **EACH** **ROW**

**BEGIN**

**SET** **NEW**.epoka = (

**SELECT** nazwa\_epoki

**FROM** epoki

**WHERE** **NEW**.rok\_wydania **BETWEEN** poczatek\_epoki **AND** koniec\_epoki

);

**END**;

FUNKCJE

1. Funkcja przeliczająca klientów o zadanym nazwisku.

**CREATE** **or** replace **FUNCTION** przelicz\_klientow(n **varchar**(80))

**RETURNS** **int**(11)

**BEGIN**

**declare** ile **int**;

**select** **count**(\*) **into** ile **from** klienci **where** nazwisko **like**

concat('%',n,'%');

**RETURN** ile;

**END**;

1. Funkcja przeliczająca ile książek zostało wydanych w zadanym roku.

**CREATE** **or** replace **FUNCTION** przelicz\_rok\_wydania(m **int**)

**RETURNS** **int**(11)

**BEGIN**

**declare** ile **int**;

**select** **count**(\*) **into** ile **from** ksiazki\_info **where** rok\_wydania **like**

concat('%',m,'%');

**RETURN** ile;

**END**;

1. Funkcja przeliczająca średnią ocenę książki.

**CREATE** **or** replace **FUNCTION** srednia\_ocena\_ksiazki(ksiazka\_id **INT**)

**RETURNS** **FLOAT**

**BEGIN**

**DECLARE** srednia\_ocena **FLOAT**;

**SELECT** **AVG**(ocena) **INTO** srednia\_ocena

**FROM** ksiazki\_oceny

**WHERE** id\_ksiazki = ksiazka\_id;

**RETURN** **COALESCE**(srednia\_ocena, 0);

**end**;

PROCEDURY

1. Procedura wyświetlająca tylko klientów o zadanym nazwisku.

**CREATE** **or** replace **PROCEDURE** wyswietl\_klientow (**IN** k **varchar**(80),

**OUT** ile **int**)

**BEGIN**

**select** imie, nazwisko **from** klienci **where** nazwisko = k;

**set** ile = FOUND\_ROWS();

**end**;

1. Procedura wyświetlająca klientów, którzy nie kupili nic przez 30 dni od dnia dzisiejszego.

**CREATE** **or** replace **PROCEDURE** nieaktywni\_klienci()

**BEGIN**

**SELECT**

**DISTINCT** klienci.id\_klient,

klienci.email

**FROM** klienci

**LEFT** **JOIN** zamowienia **ON** klienci.id\_klient = zamowienia.id\_klient

**WHERE** zamowienia.data\_zamowienia **IS** **NULL** **OR** zamowienia.data\_zamowienia < DATE\_SUB(CURDATE(), **INTERVAL** 30 **DAY**);

**end**;

1. Procedura przydzielająca zniżkę w zależności od tego, czy klient kupił kiedyś którąś z książek „Harry Potter” i dostosowująca treść maila.

**CREATE** **or** replace **PROCEDURE** znizka()

**BEGIN**

**SELECT**

klienci.id\_klient,

klienci.email,

**CASE**

**WHEN** **EXISTS** (**SELECT** 1 **FROM** zamowienia

**JOIN** koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** klienci.id\_klient = zamowienia.id\_klient

**AND** ksiazki\_info.tytul **LIKE** "%Harry Potter%") **THEN** "Wróć do Hogwartu!!"

**WHEN** **EXISTS** (**SELECT** 1 **FROM** zamowienia

**JOIN** koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** klienci.id\_klient = zamowienia.id\_klient

**AND** ksiazki\_info.tytul **NOT** **LIKE** "%Harry Potter%") **THEN** "Dołącz do uczniów Hogwartu i odkryj fascynujący świat czarów i magii"

**ELSE** "Witaj mugolu!"

**END** **AS** tresc\_wiadomosci,

**CASE**

**WHEN** **EXISTS** (**SELECT** 1 **FROM** zamowienia

**JOIN** koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** klienci.id\_klient = zamowienia.id\_klient

**AND** ksiazki\_info.tytul **LIKE** "%Harry Potter%") **THEN** "BACKTOHARRY"

**WHEN** **EXISTS** (**SELECT** 1 **FROM** zamowienia

**JOIN** koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** klienci.id\_klient = zamowienia.id\_klient

**AND** ksiazki\_info.tytul **NOT** **LIKE** "%Harry Potter%") **THEN** "DISCOVERHARRY"

**ELSE** "FIRSTTIME"

**END** **AS** kupon,

**CASE**

**WHEN** **EXISTS** (**SELECT** 1 **FROM** zamowienia

**JOIN** koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** klienci.id\_klient = zamowienia.id\_klient

**AND** ksiazki\_info.tytul **LIKE** "%Harry Potter%") **THEN** "10%"

**WHEN** **EXISTS** (**SELECT** 1 **FROM** zamowienia

**JOIN** koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN** ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**WHERE** klienci.id\_klient = zamowienia.id\_klient

**AND** ksiazki\_info.tytul **NOT** **LIKE** "%Harry Potter%") **THEN** "15%"

**ELSE** "25%"

**END** **AS** przyslugujaca\_znizka

**FROM** klienci;

**end**;

WIDOKI

1. Widok pokazujący historię zamówień.

**CREATE** **or** replace **VIEW** vhistoria\_zamowien **AS**

**SELECT**

zamowienia.id\_zamowienia,

zamowienia.data\_zamowienia,

klienci.imie,

klienci.nazwisko,

ksiazki\_info.tytul **AS** tytul\_ksiazki,

status\_dostawy.status **AS** status\_zamowienia

**FROM**

zamowienia

**JOIN**

koszyk **ON** zamowienia.id\_zamowienia = koszyk.id\_zamowienia

**JOIN**

ksiazki\_info **ON** koszyk.id\_ksiazki = ksiazki\_info.id\_ksiazki

**JOIN**

status\_dostawy **ON** zamowienia.id\_status = status\_dostawy.id\_status

**JOIN**

klienci **ON** zamowienia.id\_klient = klienci.id\_klient;

1. Widok pokazujący 10 klientów, którzy najwięcej kupili.

**CREATE** **or** replace **VIEW** vtop\_klientow **AS**

**SELECT**

klienci.id\_klient,

klienci.imie,

klienci.nazwisko,

**COUNT**(zamowienia.id\_zamowienia) **AS** liczba\_zamowien

**FROM**

klienci

**LEFT** **JOIN**

zamowienia **ON** klienci.id\_klient = zamowienia.id\_klient

**GROUP** **BY**

klienci.id\_klient

**ORDER** **BY**

liczba\_zamowien **DESC**

**LIMIT** 10;

1. Widok pokazujący dostępne w magazynie książki.

**create** **or** replace **VIEW** vdostepne\_ksiazki **AS**

**SELECT**

ksiazki\_info.id\_ksiazki,

ksiazki\_info.tytul,

magazyn.stan\_magazynowy

**FROM**

ksiazki\_info

**LEFT** **JOIN**

magazyn **ON** ksiazki\_info.id\_ksiazki = magazyn.id\_ksiazki

**WHERE**

magazyn.hurtownia = 0 **AND** magazyn.stan\_magazynowy > 0;