**UNIWERSYTET RZESZOWSKI**

**Wydział: Matematyczno-przyrodniczy**

**Kierunek: Informatyka**

**Hurtownie Danych.**

Hurtownia dotycząca sprzedaży książek w księgarni.

**Wykonał**

Adrian Szuptarski

Informatyka UR

Rok IV

RZESZÓW 2017.01.25

Spis treści

[1. Informacje Ogólne 2](#_Toc473121494)

[2. Zastosowanie aplikacji 3](#_Toc473121495)

[3. Wymagania aplikacji 3](#_Toc473121496)

[4. Architektura oprogramowania 4](#_Toc473121497)

[5. Technologie programistyczne 4](#_Toc473121498)

[6. Struktura bazy danych 6](#_Toc473121499)

# Informacje Ogólne

Opracowana aplikacja służy do eksploracji hurtowni danych dotyczącej sprzedaży książek w poszczególnych księgarniach. Korzystając z biblioteki jar „weka” w wersji 3\_8 oraz bazy danych w formacie Postgress, która łączy się do programu za pomocą konektora PostgreSQL w wersji 9.4.1207 aplikacja jest w stanie na załączonej bazie danych wyliczyć statystyki, agregować dane oraz oceniać jakość klasyfikatora

Aplikacja łącząc się z baza danych „Ksiegarnie” korzysta z portu 5432, nazwy użytkownika „postgresql” oraz hasła „szupek”.

Okienkowy interfejs ułatwia komunikację z użytkownikiem, który ma do dyspozycji 2 zakładki:

- „Agregacja: zwijanie” do agregacji danych pobranych do tabeli za pomocą polecenia bazodanowego sql z wykorzystaniem funkcji rollup. Polecenie uzyte w tej zakładce pokazuje nam nazwę księgarni, region w jakim księgarnia się znajduje, miasto w jakim jest księgarnia, oraz zlicza wszystkie transakcje dokonane przez klientów we wszystkich księgarniach. Agregacja użyta w tej zakładce pozwala zawęzić poszukiwania do poszczególnych miast. Wtedy polecenie SQL wyszukuje wszystkie transakcje dokonane w wybranym mieście.

-Statystyki – do obliczania statystyk na bazie danych. Statystyki dotyczą dwóch tabel, Zakupy- gdzie aplikacja pokazuje nam kiedy dokonano pierwszych zakupów, a kiedy dokonano ostatnich zakupów. Oraz statystyki na tabeli książki, gdzie aplikacja pokazuje nam w którym roku napisano pierwszą i ostatnią książkę.

-klasyfikacja – w zakakładce klasyfikacje mamy do dyspozycji algorytm kNN, który dzieli tablice wejściową na dwa podzbiory - dane treningowe stanowią 70% tablicy, zaś testowe 30% , i przy pomocy wprowadzonej liczbie najbliższych sąsiadów oraz metryce pozwala wyliczyć jak miasto, w którym żyje klient wpływa na decyzje o zakupie danego gatunku oraz wydawnictwa książki.

# Zastosowanie aplikacji

Aplikacja ma służyć do zarządzania księgarniami. Głównym elementem programu jest przechowywanie danych dotyczących sprzedaży książek w poszczególnych księgarniach. Każda transakcja ma przypisanego klienta, księgarnie oraz towar jaki został zakupiony.

Wszystkie te informacje posłużą do dalszej funkcjonalności programu w celu agregacji oraz filtrowania danych przez użyte algorytmu.

# Struktura programu

W programie zastosowano dwie funkcja. Pierwsza, która służy do agregacji zapytania, w którym zapytanie wyświetlające wszystkie transakcje dla danej księgarni i w każdej miejscowości, można zagregować do zapytania wyświetlającego wszystkie transakcje, wszystkich księgarni w danej miejscowości.

Druga funkcja służy do klasyfikowania kNN. Do wybory w tej funkcji mamy wybór metryki wykorzystywanej, oraz ilość sąsiadów. Algorytm ten przewiduje jak wydawnictwo książki oraz tytuł wpływa na miejscowość klientów zamieszkujących dane miasto

# Wymagania aplikacji

Program do poprawnego funkcjonowania wymaga zainstalowanej platforma Java oraz aktywnej bazy danych w programie PostreSQL.

Zalecane jest też minimum 100MB wolnego miejsca na dysku oraz komputer klasy core i3 pierwszej generacji lub wyżej.

# Architektura oprogramowania

Baza danych użyta do aplikacji została stworzona w języku SQL za pomocą oprogramowania postgreSQL. Baza posiada 5 tabel: zakupy, daty, książki, klienci, księgarnie. Dane do bazy zostały wygenerowane w sposób losowy za pomocą oprogramowania Excel, a następnie zaimportowane do postgreSQL w formacie CSV rozdzielonym przecinkami. Tabela zakupy posiada 102 rekordy dokonanych transakcji.

Aplikacja została napisana w języku oprogramowania Java za pomocą kompilatora NetBeans. Do stworzenia aplikacji okienkowej użyto technologii JavaFx, zaś edycja pliku xml z interfejsem odbywała się za pomocą oprogramowania JAVA FX scene builder 2.0

Aplikacja składa się z 5 klas odpowiedzialnych m.i za łączenie z bazą danych (db.java), wyświetlanie danych (FXMLController.java) , przesyłanie do użytkownika ostrzeżeń (utils.java), stworzenia modelu danych – klasy księgarnie i niezbędnych do niej konstruktorów, geterów i seterów (drogerie.java)

# Technologie programistyczne

**Java** – obiektowy język programowania stworzony przez grupę roboczą pod kierunkiem Jamesa Goslinga z firmy Sun Microsystems. Java jest językiem tworzenia programów źródłowych kompilowanych do kodu bajtowego, czyli postaci wykonywanej przez maszynę wirtualną. Język cechuje się silnym typowaniem. Jego podstawowe koncepcje zostały przejęte z języka Smalltalk (maszyna wirtualna, zarządzanie pamięcią) oraz z języka C++ (duża część składni i słów kluczowych).

**PostgreSQL**, obok MySQL i Firebird, stanowi jeden z trzech najpopularniejszych wolnodostępnych systemów zarządzania relacyjnymi bazami danych. Początkowo opracowywany na Uniwersytecie Kalifornijskim w Berkeley i opublikowany pod nazwą Ingres. W miarę rozwoju i zwiększania funkcjonalności, baza danych otrzymała nazwy Postgres95 i ostatecznie PostgreSQL, aby upamiętnić pierwowzór oraz zaznaczyć zgodność ze standardem SQL.

PostgreSQL zalicza się do baz typu RDBMS z rozszerzeniami obiektowymi. W samej bazie można pisać procedury składowane w różnych językach programowania:

-wbudowany język PL/pgSQL podobny do proceduralnego języka PL/SQL w bazie Oracle,

- języki skryptowe m.in. PL/LOLCODE, PL/Lua, PL/Perl, pl/PHP, PL/Python, PL/Ruby, PL/Tcl, PL/Scheme, języki kompilowane C, C++, lub Java (jako PL/Java, język statystyczny R jako PL/R

W PostgreSQL zaimplementowano obsługę wielu typów indeksów takich jak B-drzewo, Hash, R–drzewo i GiST.

PostgreSQL ma zaimplementowany mechanizm MVCC (Multiversion Concurrency Control) do zarządzania transakcjami. Mechanizm ten umożliwia udostępnienie tej samej krotki dla więcej niż jednej transakcji. Równocześnie może istnieć przynajmniej kilka wersji tej samej krotki, które nie są widoczne dla innych użytkowników do zakończenia danych transakcji. Dzięki temu baza danych wydajnie zachowuje zasadę ACID.

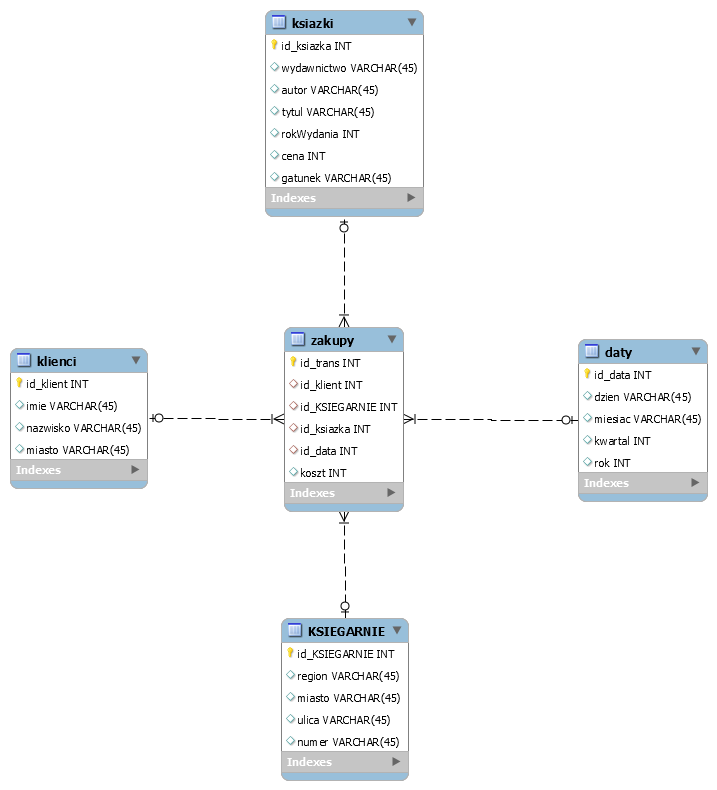
Reguły są to elementy aktywne, które mogą być stosowane m.in. jako rozszerzenie widoków. Za pomocą reguł można zrealizować widoki modyfikujące, jako że reguły mogą mieć w definicji klauzulę INSTEAD OF, co spowoduje, że robiąc zapytanie INSERT wstawiające dane do widoku, reguła zamiast tego zapytania zrobi inne, które wstawi dane do tabeli (bądź wielu tabel), z której widok odczytuje dane.

**SQL** (Structured Query Language) jest najbardziej znanym i rozpowszechnionym strukturalnym językiem zapytań, który służy do tworzenia, modyfikacji oraz zarządzania bazami danych. SQL należy do języków deklaratywnych. Oznacza to, że jest on oparty na paradygmatach programowania nie opartych na programowaniu imperatywnym, które opisuje procesy wykonywania jako sekwencję instrukcji zmieniających stan programu. Sposób przechowywania i pobierania danych jest zależny od decyzji systemu zarządzania bazą danych (Database Management System, DBMS).

Knn - Jedną z najistotniejszych nieparametrycznych metod klasyfikacji jest metoda K-najbliższych sąsiadów, oznaczana przez K-NN. W metodzie tej zaliczamy rozpoznawany obiekt do tej klasy, do której należy większość z jego K najbliższych sąsiadów. Estymator prawdopodobieństwa a posteriori przynależności obserwacji x do klasy k jest wyliczany jako udział obserwacji z tej klasy wśród jej K najbliższych sąsiadów

- Agregacja jak sama nazwa wskazuje służy do agregacji danych pobranych do  
tabeli za pomocą polecenia bazodanowego SQL z wykorzystaniem funkcji agregującej  
rollup,

# Struktura bazy danych



Baza danych składa się z 5 tabel, w których są przechowywane następujące informacje:

Tabela daty- przechowuje informacje dotyczące dat poszczególnych transakcji. W tabeli znajdują się takie pola jak: dzień, miesiąc, rok, kwartał, id\_data

Tabela klienci - przechowuje informacje dotyczące klientów dokonujących transakcji w danej księgarni. W tabeli znajdują się następujące informacje: imie, nazwisko, miasto, id\_klient

Tabela książki - zawiera informacje dotyczące książek występujących w transakcjach. Tabela ta zawiera następujące pola: nazwa, tytuł, wydawnictwo, gatunek, rok wydania, id\_ksiazka

Tabela księgarnie- zawiera informacje dotyczące poszczególnych księgarni. Tabela zawiera następujące pola: region, miasto, ulica, numer nazwa, id\_siegarni

Tabela zakupy - jest to tabela faktów przechowująca informacje dotyczące transakcji dokonywanych w poszczególnych księgarniach przez poszczególnych klientów. Tabela ta ma następujące pola: id\_transakcji, id\_klient, id\_ksiegarni, id\_data, id\_ksiazki, koszt. Tabela ta jest centralna tabelą łączącą wszystkie tabele.

Poniżej przedstawiam zapytania wykorzystywane w moim programie.

W pierwszej funkcjonalności mojego programu wykorzystałem funkcję agregacji przy poniższym zapytaniu:

select ks.nazwa, ks.region, ks.miasto, count(\*)\n

from ksiegarnie ks, klienci kl, zakupy z

where z.id\_ksiegarnie=ks.id\_ksiegarnie and ks.miasto="str"+

group by rollup(ks.nazwa, ks.region,ks.miasto)

Zapytanie to wyświetla nazwę książki, region, oraz miasto księgarni w jakiej dokonano zakupu. Zapytanie zlicza ile transakcji wykonane jest w danej księgarni, w księgarniach z danego regionu, oraz w danej księgarni z danego regionu dla poszczególnych miast, oraz sumę wszystkich operacji.

Gdzie parametr **str** jest wykorzystywany do zawężenia wyszukiwani do wyszukiwania transakcji dla księgarni w wybranym mieście.

Do funkcji kNN wykorzystałem następujące zapytanie SQL:

select ks.wydawnictwo,ks.gatunek, kl.miasto

from zakupy z,ksiazki ks,klienci kl

where ks.id\_ksiazka=z.id\_ksiazka and kl.id\_klient=z.id\_klient;

Zapytanie to wyświetla wydawnictwo, oraz gatunek książki i miasto osoby, która kupiła książkę. Algorytm kNN jako decyzje wykorzystuje pole miasto, a więc algorytm szacuje jak wydawnictwo, oraz gatunek książki wpływają na miasto, w których mieszkają klienci.