



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

Temat Projektu:

**“Porównanie środowisk bazodanowych MySQL pod kątem
operacji wykonywanych na bazie danych”**

Stachiewicz Dawid 173218, Sulisz Szymon 173221

Opiekun pracy:

Mgr. Inż. Dariusz Rączka

Rzeszów, 2024

Spis treści

1. Wstęp	1
1.1 Wprowadzenie do tematu.	1
1.2 Cel i zakres projektu:	1
2. Rozwinięcie	2
2.1 SQL Server Management Studio	2
2.2 MySQL – MariaDB	3
2.3 Schemat bazy danych wykorzystanej w projekcie.....	5
2.3.1 Dane przechowywane w poszczególnych tabelach	7
2.4 Import danych	11
2.4.1 Dla SSMS.....	12
2.4.2 Dla MariaDB.....	14
2.5 Wnioski z procesu importowania bazy danych	15
2.5.1 SSMS	15
2.5.2 MariaDB	15
3. Porównanie pracy obu serwerów podczas ich obciążania.....	16
3.1 Polecenia jakim baza danych była obciążana	16
3.2 Porównanie wyników podczas obciążania serwera	18
3.2.1 Sposób zbierania danych na temat obciążenia.....	18
3.2.2 Porównanie wyników wydajnościowych.....	19
3.2.3 Podsumowanie	23
4. Bibliografia.....	24
5. Spis rysunków.....	25
6. Spis wykresów	26

1. Wstęp

1.1 Wprowadzenie do tematu.

W dzisiejszym dynamicznym świecie informatyki, w którym dane stanowią kluczowy zasób, wybór odpowiedniego środowiska bazodanowego jest niezwykle istotny dla efektywnego zarządzania informacjami oraz sprawnego działania aplikacji. W kontekście projektu dotyczącego administracji systemów rozproszonych, badanie i porównanie dwóch popularnych platform - SQL Server Management Studio oraz MySQL-MariaDB - staje się kluczowym krokiem w procesie zrozumienia ich charakterystyk, możliwości i ograniczeń. Przeprowadzenie takiego porównania pozwala zidentyfikować różnice między tymi rozwiązaniami oraz określić, które z nich lepiej odpowiadają na konkretne potrzeby i wymagania projektów rozproszonych.



Rysunek 1 Środowisko bazodanowe MySQL

1.2 Cel i zakres projektu:

Celem niniejszej pracy jest przeprowadzenie kompleksowego porównania obu tych środowisk z uwzględnieniem ich technicznych aspektów, funkcjonalności oraz wydajności w kontekście istniejącej bazy danych. Projekt rozpoczyna się od importowania istniejącej bazy danych do obu środowisk, zapewniając spójność danych i struktury. Następnie przeprowadzane są testy wydajnościowe, które obejmują różnorodne scenariusze obciążeniowe, zapytania i operacje na danych. Testy te mają na celu ocenę wydajności, skalowalności i stabilności obu platform w praktycznych warunkach. Ostatecznie, wyniki testów są analizowane i porównywane, co umożliwia wyciągnięcie wniosków dotyczących

efektywności i przydatności w kontekście konkretnej aplikacji lub projektu rozproszonego. Poprzez to podejście projektowe możliwe będzie lepsze zrozumienie różnic między MySQL Server Management Studio a MariaDB oraz wybór optymalnego rozwiązania dla konkretnych potrzeb projektowych.

2. Rozwinięcie

2.1 SQL Server Management Studio

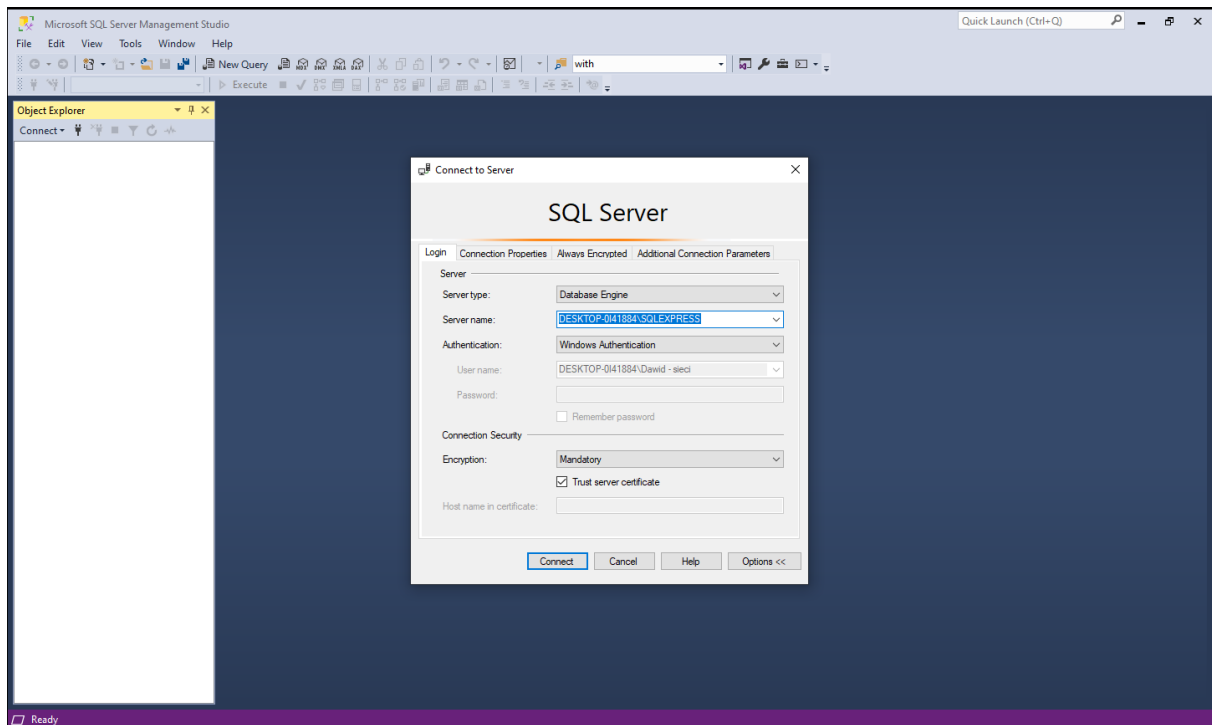
SQL Server Management Studio (SSMS) to narzędzie stworzone przez firmę Microsoft, zaprojektowane do zarządzania bazami danych działającymi na platformie Microsoft SQL Server. Jest to kompleksowe środowisko do zarządzania bazami danych, które umożliwia administratorom i programistom wykonywanie szeregu operacji administracyjnych, jak również rozwijanie, monitorowanie i optymalizację baz danych.

SSMS umożliwia wykonywanie różnorodnych zadań, takich jak:

- Tworzenie i edycja baz danych oraz tabel.
- Wykonywanie zapytań SQL do przeglądania, edycji i analizowania danych.
- Zarządzanie uprawnieniami dostępu do bazy danych.
- Tworzenie i zarządzanie procedurami składowanymi, funkcjami, wyzwalaczami i widokami.
- Monitorowanie wydajności bazy danych poprzez analizę planów wykonania zapytań, raporty o obciążeniu systemu itp.
- Wykonywanie operacji kopii zapasowych i przywracania danych.
- Konfigurowanie i monitorowanie replikacji danych.
- Rozwiązywanie problemów i diagnozowanie błędów w bazie danych.

My korzystamy z SSMS głównie w celu zarządzania bazą danych SQL Server, importowania danych, przeprowadzania zapytań SQL oraz monitorowania i optymalizowania wydajności bazy danych.

Aby rozpocząć korzystanie z SSMS, najpierw pobraliśmy aplikację odwiedzając [oficjalną stronę internetową firmy Microsoft](#) i wybierając odpowiednią wersję SSMS zgodną z naszym systemem operacyjnym jakim jest Windows 10 Pro. Po pobraniu pliku instalacyjnego, postępowaliśmy zgodnie z krokami instalacyjnymi.



Rysunek 2 Zrzut ekranu skonfigurowanego serwera MySQL

2.2 MySQL – MariaDB

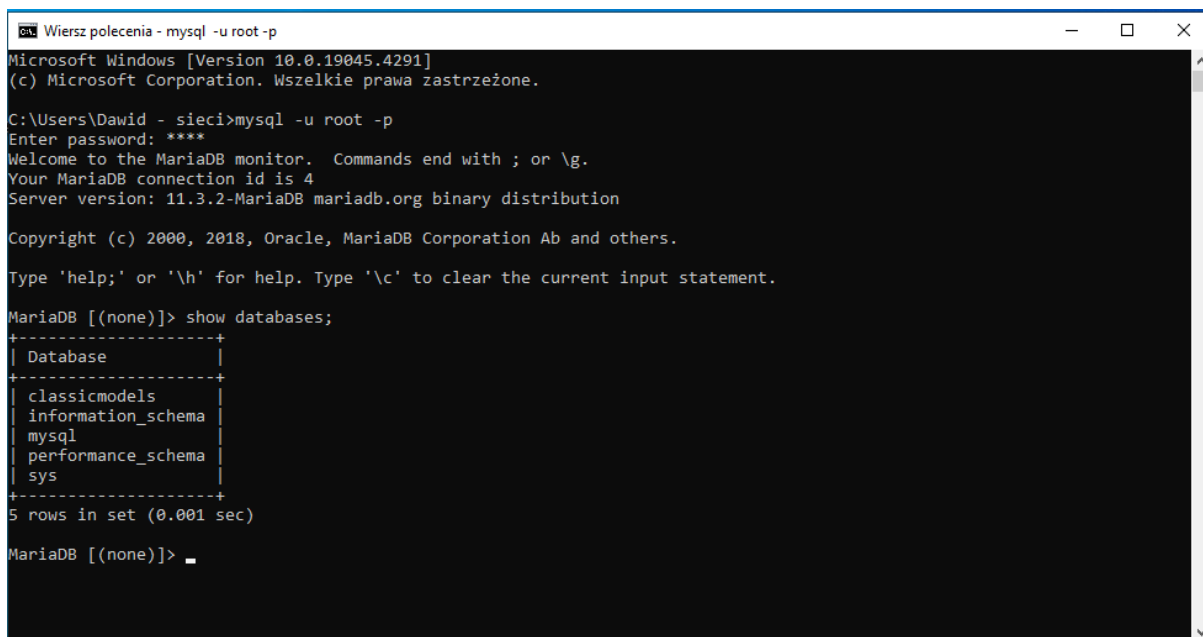
MySQL i MariaDB to otwarte źródłowe systemy zarządzania relacyjnymi bazami danych (RDBMS), które cieszą się ogromną popularnością wśród deweloperów i przedsiębiorstw. MariaDB jest forkem MySQLa, co oznacza, że bazuje na kodzie źródłowym MySQLa, ale także wprowadza własne ulepszenia i nowe funkcje.

Te systemy bazodanowe są wykorzystywane do przechowywania, zarządzania i dostępu do danych w sposób zorganizowany, relacyjny i bezpieczny. Mogą być stosowane w różnych aplikacjach i scenariuszach, od małych witryn internetowych po duże aplikacje przedsiębiorstwowe.

MySQLa/MariaDB wykorzystujemy głównie do:

- Przechowywania danych: Dzięki relacyjnej strukturze danych, MySQL/MariaDB są idealnym rozwiązaniem do przechowywania danych w tabelach, zapewniając łatwy dostęp i wysoką wydajność.

- Zarządzania danymi: Oferują zaawansowane narzędzia do tworzenia, modyfikowania i usuwania danych, zapewniając pełną kontrolę nad informacjami przechowywanymi w bazie danych.
- Wykonywania zapytań: Pozwalają na wykonywanie złożonych zapytań SQL w celu przeglądania, aktualizacji, wstawiania i usuwania danych z bazy danych.
- Skalowania aplikacji: MySQL/MariaDB są elastyczne i łatwe w skalowaniu, co pozwala na obsługę rosnącej liczby użytkowników i danych w aplikacji.
- Zapewniania bezpieczeństwa: Oferują mechanizmy autoryzacji, uwierzytelniania i kontroli dostępu, aby chronić dane przed nieautoryzowanym dostępem i zagrożeniami bezpieczeństwa.
- Optymalizacji wydajności: Posiadają narzędzia do optymalizacji wydajności, takie jak indeksowanie danych, partycjonowanie tabel, analiza planów wykonania zapytań itp., co pozwala na uzyskanie szybkiego dostępu do danych nawet w przypadku dużych zbiorów danych.



```

Wiersz polecenia - mysql -u root -p
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Dawid - sieci>mysql -u root -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 11.3.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| classicmodels |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]>

```

Rysunek 3 Zrzut ekranu konsoli MySQL - MariaDB

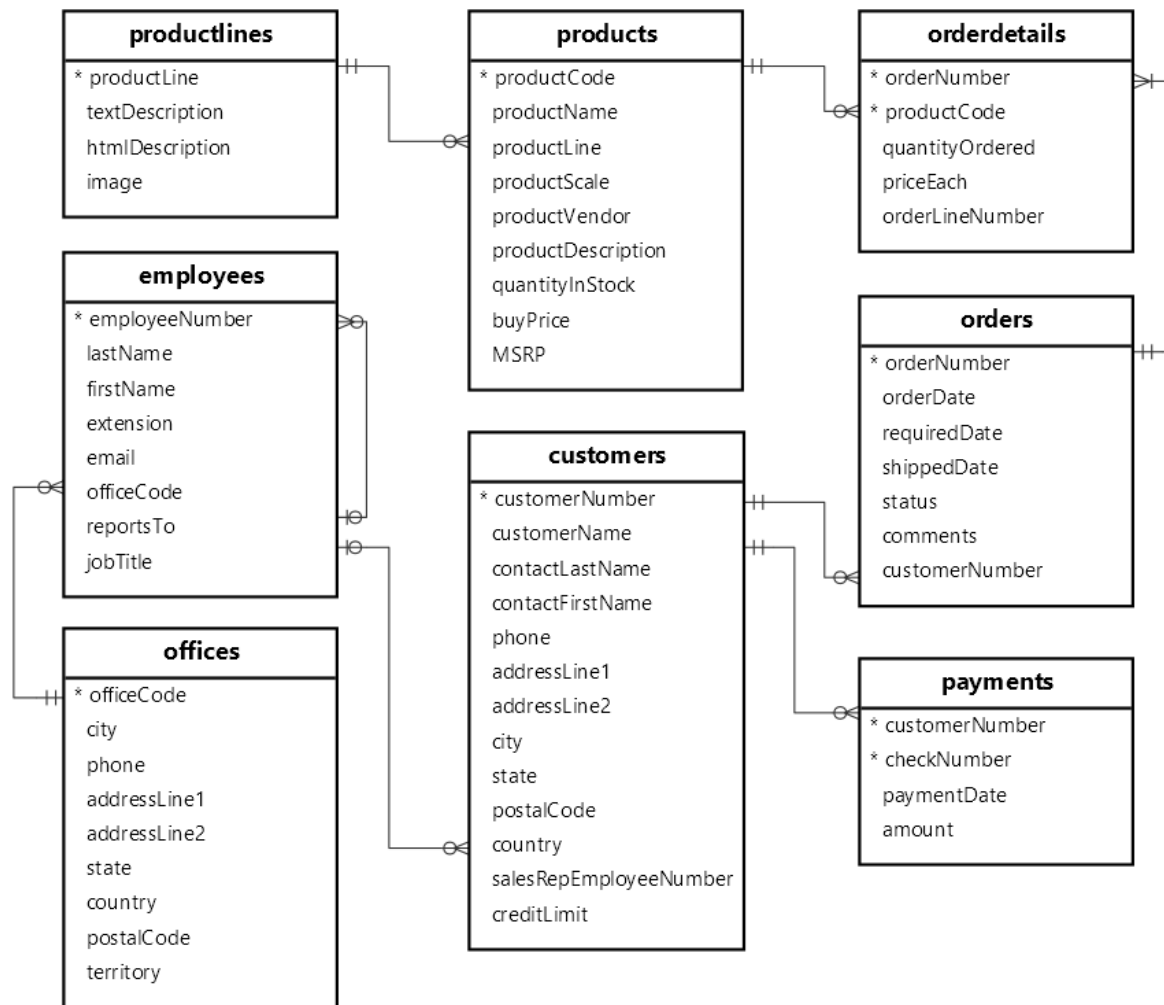
2.3 Schemat bazy danych wykorzystanej w projekcie.

W naszym projekcie używamy bazy danych „classicmodels” jako przykładowej bazy danych MySQL. Baza danych classicmodels opisuje sprzedaż detaliczną na skalę klasycznych samochodów. Zawiera typowe dane biznesowe, w tym informacje o klientach, produktach, zamówieniach sprzedaży, pozycjach zamówienia sprzedaży i wiele innych.

Schemat tej bazy danych składa się z następujących tabel:

- 1) **customers:** przechowuje dane klientów.
- 2) **products:** przechowuje listę modeli samochodów w skali.
- 3) **productlines:** przechowuje listę linii produktów.
- 4) **orders:** przechowuje zamówienia sprzedaży składane przez klientów.
- 5) **orderdetails:** przechowuje pozycje zamówienia sprzedaży dla każdego zamówienia.
- 6) **payments:** przechowuje płatności dokonywane przez klientów na podstawie ich kont.
- 7) **employees:** przechowuje informacje o pracownikach oraz strukturę organizacyjną, np. kto jest podwładnym kogo.
- 8) **offices:** przechowuje dane o biurach sprzedaży.

Poniższy rysunek przedstawia diagram ER wykorzystywanej bazy:



Rysunek 4 Diagram ER bazy danych "classicmodels"

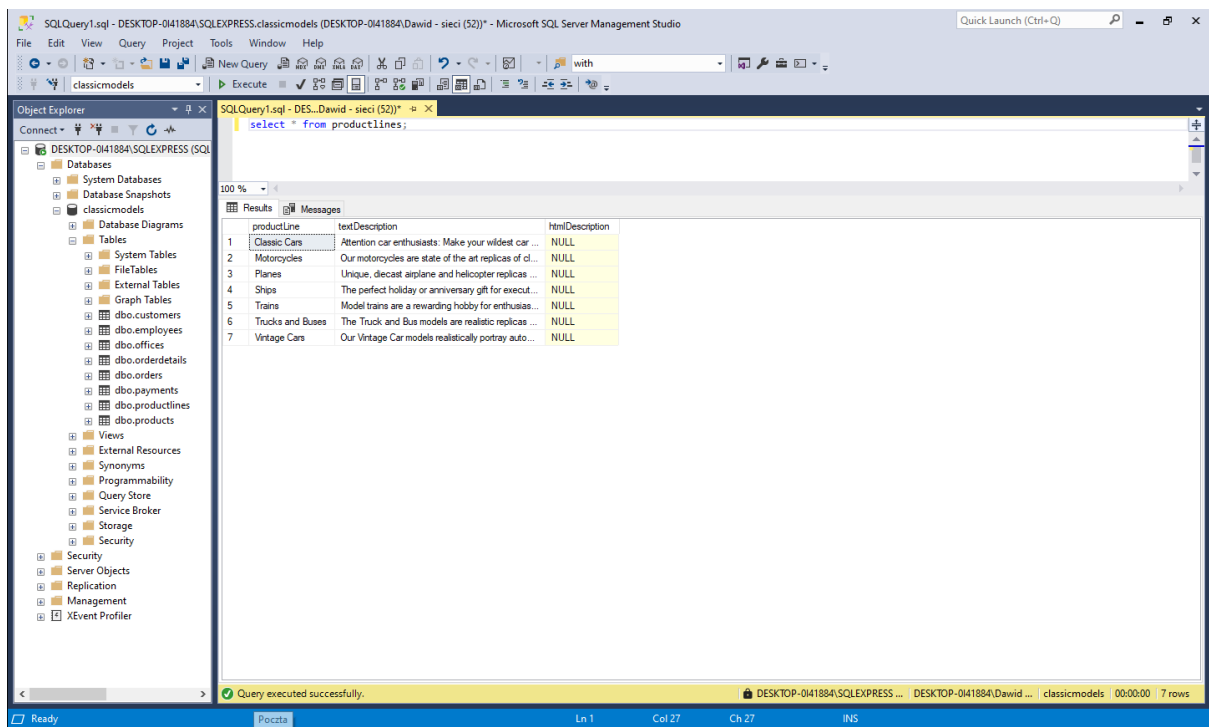
2.3.1 Dane przechowywane w poszczególnych tabelach

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit
103	Atelier graphique	Schmitt	Carine	40 32 2555	54, rue Royale	NULL	Nantes	NULL	44000	France	1370	21000.00
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.	NULL	Las Vegas	NV	83030	USA	1166	71800.00
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555	636 St Kilda Road	Level 3	Melbourne	Victoria	3004	Australia	1611	117300.00
119	La Rochelle Gifts	Labruno	Janine	40 67 8555	67, rue des Cinquante Otages	NULL	Nantes	NULL	44000	France	1370	118200.00
121	Baane Mini Imports	Bergulsen	Jonas	07-98 9555	Erling Skakkes gate 78	NULL	Slavem	NULL	4110	Norway	1504	81700.00
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450	5677 Strong St.	NULL	San Rafael	CA	97562	USA	1165	210500.00
125	Havel & Zbysek	Piestrzeniewicz	Zbysek	(26) 642-7555	ul. Filtrawa 38	NULL	Warszawa	NULL	01-012	Poland	NULL	0.00
128	Bauer See Auto. Co.	Kettel	Roland	+49 69 66 90 2555	Lyonenstr. 34	NULL	Frankfurt	NULL	60528	Germany	1504	59700.00
129	Mini Wheels Co.	Murphy	Julie	6505555787	5557 North Pendale Street	NULL	San Francisco	CA	94217	USA	1165	64600.00
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue	NULL	NYC	NY	10022	USA	1323	114900.00
141	Euro-Shopping Channel	Freyre	Diego	(91) 555 94 44	C/ Moralezaral, 86	NULL	Madrid	NULL	28034	Spain	1370	227600.00
144	Volvro Model Replicas, Co	Berglund	Christina	0921-12 3555	Berguvaggen 8	NULL	Lulea	NULL	S-958 22	Sweden	1504	53100.00
145	Danish Wholesale Imports	Petersen	Jytte	31 12 3555	Vrlobatet 34	NULL	Kobenhavn	NULL	1734	Denmark	1401	83400.00
146	Saveley & Henriot, Co.	Saveley	Mary	78 32 5555	2, rue du Commerce	NULL	Lyon	NULL	69004	France	1337	123900.00
148	Dragon Souvenirs, Ltd.	Natividad	Eric	+65 221 7555	Bronz Sok.	NULL	Singapore	NULL	079903	Singapore	1621	103800.00
151	Muscle Machine Inc.	Young	Jeff	2125557413	4092 Furth Circle	Suite 400	NYC	NY	10022	USA	1286	138500.00
157	Decast Classics Inc.	Leong	Kevin	2155551555	7586 Pompton St.	NULL	Alentown	PA	70267	USA	1216	100600.00
161	Technics Stores Inc.	Hashimoto	Juri	6505556809	9408 Furth Circle	NULL	Burlingame	CA	94217	USA	1165	84500.00
166	Handi Gifts Co	Victorino	Wendy	+65 224 1555	105 Linden Road Sandown	2nd Floor	Singapore	NULL	069045	Singapore	1612	97900.00
167	Herkku Gifts Co	Oestari	Yvesel	+47 2267 3215	Behnmen St. 121	PR 334 Se...	Bergen	NULL	N 5804	Norway	1504	96800.00
168	American Souvenirs Inc.	Franco	Kath	2035557845	149 Spinnaker Dr.	Suite 101	New Haven	CT	97823	USA	1286	0.00
169	Porto Imports Co	de Castro	Isabel	(1) 366-5555	Estrada da saude n. 58	NULL	Lisboa	NULL	1756	Portugal	NULL	0.00
171	Decadive Designs Imports	Rincol	Marlene	20 16 1555	184, chaussee de Tournai	NULL	Lille	NULL	59000	France	1370	92900.00
172	La Corne D'abondance...	Bertrand	Marie	(1) 42 34 2555	265, boulevard de Chaurone	NULL	Paris	NULL	75012	France	1337	94300.00
173	Cambridge Collectables ...	Tseng	Jenny	6175555555	4658 Baden Av.	NULL	Cambridge	MA	51247	USA	1188	43400.00
175	Gift Depot Inc.	King	Julie	2035552570	25593 South Bay Ln.	NULL	Bridgewater	CT	97562	USA	1323	94300.00
177	Osaka Souvenirs Co.	Kentary	Mary	+81 06 6342 5555	1-6-20 Dajima	NULL	Kiaku	Osaka	530-0003	Japan	1621	81200.00
181	Vitachrome Inc.	Frick	Michael	2125551500	2678 Kingston Rd.	Suite 101	NYC	NY	10022	USA	1286	76400.00

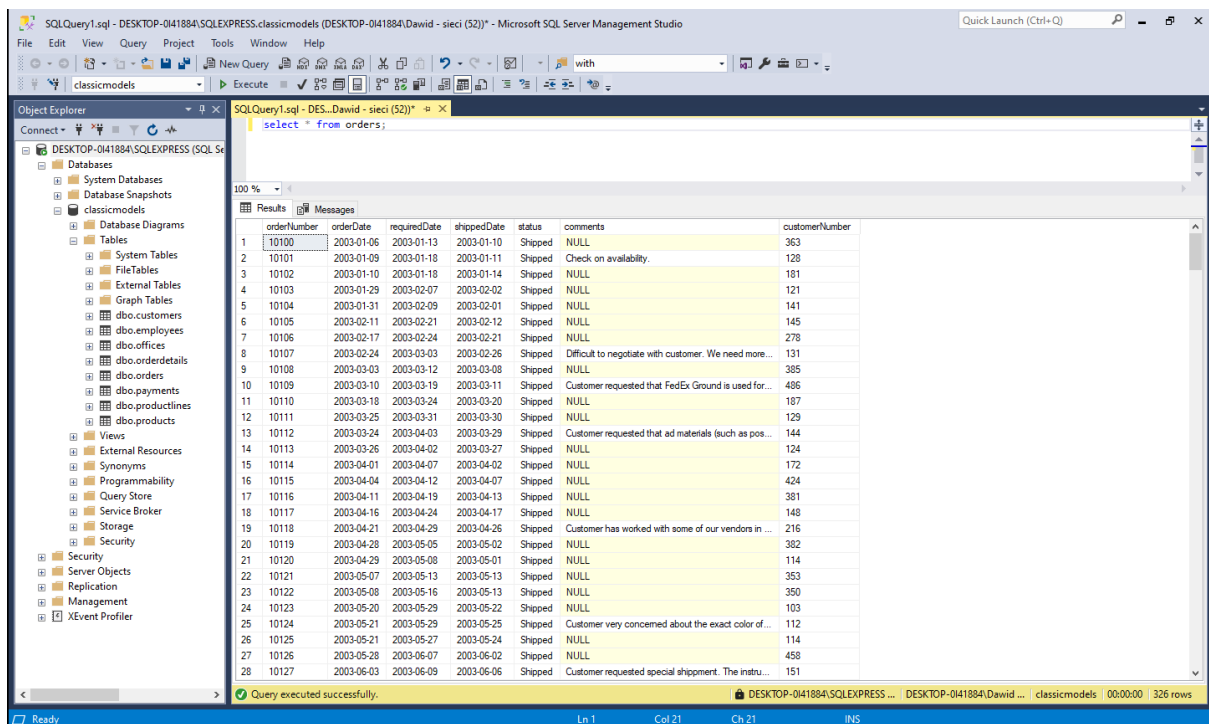
Rysunek 5 Wyświetlenie zawartości tabeli "customers"

productCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice	MSRP
S10_1678	1969 Harley Davidson Ultimate Chopper	Motorcycles	1:10	Min Lin Decast	This replica features working kickstand, front suspe...	7933	48.81	95.70
S10_1949	1952 Alpine Renault 1300	Classic Cars	1:10	Classic Metal Creations	Turnable front wheels; steering function; detailed int...	7305	98.58	214.30
S10_2016	1996 Moto Guzzi 1100	Motorcycles	1:10	Highway 66 Mini Classics	Official Moto Guzzi logos and insignias; saddle bags...	6625	68.99	118.94
S10_4638	2003 Harley-Davidson Eagle Drag Bike	Motorcycles	1:10	Red Start Decast	Model features; official Harley Davidson logos and i...	5582	91.02	193.66
S10_4757	1972 Alfa Romeo GTA	Classic Cars	1:10	Motor City Art Classics	Features include: Turnable front wheels; steering fu...	3252	85.68	136.00
S10_4962	1962 Lancia Delta 16V	Classic Cars	1:10	Second Gear Decast	Features include: Turnable front wheels; steering fu...	6791	103.42	147.74
S12_1099	1968 Ford Mustang	Classic Cars	1:12	Autoart Studio Design	Hood, doors and trunk all open to reveal highly deta...	68	95.34	194.57
S12_1108	2001 Ferrari Enzo	Classic Cars	1:12	Second Gear Decast	Turnable front wheels; steering function; detailed int...	3619	95.59	207.80
S12_1666	1958 Setra Bus	Trucks and Buses	1:12	Welly Decast Productions	Model features 30 windows, skylights & glare resista...	1579	77.90	136.67
S12_2823	2002 Suzuki XREO	Motorcycles	1:12	Unimax Art Galleries	Official logos and insignias; saddle bags located on ...	9997	66.27	150.62
S12_3148	1969 Corvair Monza	Classic Cars	1:18	Welly Decast Productions	1:18 scale die-cast about 10 1/2" long doors open, ho...	6906	89.14	151.08
S12_3380	1968 Dodge Charger	Classic Cars	1:12	Welly Decast Productions	1:12 scale model of a 1968 Dodge Charger; Hood, ...	9123	75.16	117.44
S12_3891	1969 Ford Falcon	Classic Cars	1:12	Second Gear Decast	Turnable front wheels; steering function; detailed int...	1049	83.05	173.02
S12_3990	1970 Plymouth Hemi Cuda	Classic Cars	1:12	Studio M Art Models	Very detailed 1970 Plymouth Cuda model in 1:12 sc...	5663	31.92	79.80
S12_4473	1957 Chevy Pickup	Trucks and Buses	1:12	Exoto Designs	1:12 scale die-cast about 20 1/2" long Hood opens, R...	6125	55.70	118.50
S12_4675	1969 Dodge Charger	Classic Cars	1:12	Welly Decast Productions	Detailed model of the 1969 Dodge Charger. This mo...	5873	58.33	115.16
S18_1097	1940 Ford Pickup Truck	Trucks and Buses	1:12	Studio M Art Models	This model features soft rubber tires, working steerin...	2613	58.33	116.67
S18_1129	1993 Mazda RX-7	Classic Cars	1:18	Highway 66 Mini Classics	This model features, opening hood, opening doors, ...	3975	83.51	141.54
S18_1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, and fu...	8693	60.62	102.74
S18_1367	1936 Mercedes-Benz 500K Special Roadster	Vintage Cars	1:18	Studio M Art Models	This 1:18 scale replica is constructed of heavy die-c...	8635	24.26	53.91
S18_1589	1965 Aston Martin DB5	Classic Cars	1:18	Classic Metal Creations	Die-cast model of the silver 1965 Aston Martin DB5 i...	9042	65.96	124.44
S18_1662	1980s Black Hawk Helicopter	Planes	1:18	Red Start Decast	1:18 scale replica of actual Army's UH-60L BLACK ...	5330	77.27	157.69
S18_1749	1917 Grand Touring Sedan	Vintage Cars	1:18	Welly Decast Productions	This 1:18 scale replica of the 1917 Grand Touring c...	2724	86.70	170.00
S18_1889	1948 Porsche 356-A Roadster	Classic Cars	1:18	Gearbox Collectibles	This precision die-cast replica features opening door...	8826	53.90	77.00
S18_1984	1995 Honda Civic	Classic Cars	1:18	Min Lin Decast	This model features, opening hood, opening doors, ...	9772	93.89	142.25
S18_2238	1998 Chrysler Plymouth Proville	Classic Cars	1:18	Gearbox Collectibles	Turnable front wheels; steering function; detailed int...	4724	101.51	163.73
S18_2248	1911 Ford Town Car	Vintage Cars	1:18	Motor City Art Classics	Features opening hood, opening doors, opening tru...	540	33.30	60.54
S18_2319	1964 Mercedes Tour Bus	Trucks and Buses	1:18	Unimax Art Galleries	Exact replica. 100+ parts. working steering system, ...	8258	74.86	122.73

Rysunek 6 Wyświetlenie zawartości tabeli "products"



Rysunek 7 Wyświetlenie zawartości tabeli "productlines"



Rysunek 8 Wyświetlenie zawartości tabeli "orders"

SQLQuery1.sql - DESKTOP-041884\SQLEXPRESS.classicmodels (DESKTOP-041884\David - sieci (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

classimodels Execute

Object Explorer

Connect

DESKTOP-041884\SQLEXPRESS (SQL Se

Databases

System Databases

Database Snapshots

classicmodels

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.customers

dbo.employees

dbo.offices

dbo.orderdetails

dbo.orders

dbo.payments

dbo.productlines

dbo.products

Views

External Resources

Synonyms

Programmability

Query Store

Service Broker

Storage

Security

Security

Server Objects

Replication

Management

XEEvent Profiler

SQLQuery1.sql - DESKTOP-041884\David - sieci (52)

select * from orderdetails;

100 %

Results Messages

	orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
1	10100	S18_1749	30	136.00	3
2	10100	S18_2248	50	55.09	2
3	10100	S18_4409	22	75.46	4
4	10100	S24_3969	49	35.29	1
5	10101	S18_2325	25	108.06	4
6	10101	S18_2795	26	167.06	1
7	10101	S24_1937	45	32.53	3
8	10101	S24_2022	46	44.35	2
9	10102	S18_1342	39	95.55	2
10	10102	S18_1367	41	43.13	1
11	10103	S10_1949	26	214.30	11
12	10103	S10_4962	42	119.67	4
13	10103	S12_1666	27	121.64	8
14	10103	S18_1097	35	94.50	10
15	10103	S18_2432	22	58.34	2
16	10103	S18_2949	27	92.19	12
17	10103	S18_2957	35	61.84	14
18	10103	S18_3136	25	86.92	13
19	10103	S18_3320	46	86.31	16
20	10103	S18_4600	36	98.07	5
21	10103	S18_4668	41	40.75	9
22	10103	S24_2300	36	107.34	1
23	10103	S24_4258	25	88.62	15
24	10103	S32_1268	31	92.46	3
25	10103	S32_3522	45	63.35	7
26	10103	S700_2824	42	94.07	6
27	10104	S12_3148	34	131.44	1
28	10104	S12_4473	41	111.39	9

Query executed successfully.

DESKTOP-041884\SQLEXPRESS ... DESKTOP-041884\David ... classicmodels 00:00:00 2 996 rows

Ready

Eksploator plików

Ln 1 Col 27 Ch 27 INS

Rysunek 9 Wyświetlenie zawartości tabeli "orderdetails"

SQLQuery1.sql - DESKTOP-041884\SQLEXPRESS.classicmodels (DESKTOP-041884\David - sieci (52)) - Microsoft SQL Server Management Studio

FileEditViewQueryProjectToolsWindowHelp

classmodels

Execute

with

Object Explorer

Connect

DESKTOP-041884\SQLEXPRESS (SQL Se

Databases

System Databases

Database Snapshots

classicmodels

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.customers

dbo.employees

dbo.offices

dbo.orderdetails

dbo.orders

dbo.payments

dbo.productlines

dbo.products

Views

External Resources

Synonyms

Programmability

Query Store

Service Broker

Storage

Security

Server Objects

Replication

Management

XEEvent Profiler

SQLQuery1.sql - DES_ David - sieci (52))*

select * from payments;

100 %

ResultsMessages

	customerNumber	checkNumber	paymentDate	amount
1	103	HQ336336	2004-10-19	6066.78
2	103	JM555205	2003-06-05	14571.44
3	103	OM314933	2004-12-18	1676.14
4	112	BO864823	2004-12-17	14191.12
5	112	HQ55022	2003-06-06	32641.98
6	112	ND748579	2004-08-20	33347.88
7	114	GG31455	2003-05-20	45864.03
8	114	MA765515	2004-12-15	82261.22
9	114	NP603840	2003-05-31	7565.08
10	114	NR27552	2004-03-10	44894.74
11	119	DB933704	2004-11-14	19501.82
12	119	LN373447	2004-08-08	47924.19
13	119	NG94694	2005-02-22	49523.67
14	121	DB889831	2003-02-16	50218.95
15	121	FD317790	2003-10-28	1491.38
16	121	KI831359	2004-11-04	17876.32
17	121	MA302151	2004-11-28	34638.14
18	124	AE215433	2005-03-05	101244.59
19	124	BG255406	2004-08-28	85410.87
20	124	CQ287967	2003-04-11	11044.30
21	124	ET64396	2005-04-16	83598.04
22	124	HI366474	2004-12-27	47142.70
23	124	HR86578	2004-11-02	55639.66
24	124	KI131716	2003-08-15	111654.40
25	124	LF217299	2004-03-26	43369.30
26	124	NT141748	2003-11-25	45084.38
27	128	DI925118	2003-01-28	10549.01
28	128	FA465482	2003-10-18	24101.81

Query executed successfully.

DESKTOP-041884\SQLEXPRESS ... DESKTOP-041884\David ... classicmodels 00:00:00 273 rows

Rysunek 10 Wyświetlenie zawartości tabeli "payments"

SQLQuery1.sql - DESKTOP-041884\SQLEXPRESS:classicmodels (DESKTOP-041884\David - sieci (52)) - Microsoft SQL Server Management Studio

Object Explorer: classicmodels

Query: select * from employees;

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1	Murphy	Dane	x5800	dmurphy@classicmodelcars.com	1	NULL	President
2	Patterson	Mary	x4611	mpatterson@classicmodelcars.com	1	1002	VP Sales
3	Firelli	Jeff	x9273	jfirelli@classicmodelcars.com	1	1002	VP Marketing
4	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
5	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sales Manager (EMEA)
6	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)
7	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep
8	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep
9	Firelli	Julie	x2173	jfirelli@classicmodelcars.com	2	1143	Sales Rep
10	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep
11	Tseng	Foon Yue	x2248	ftseng@classicmodelcars.com	3	1143	Sales Rep
12	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep
13	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep
14	Hernandez	Gerard	x2028	ghernandez@classicmodelcars.com	4	1102	Sales Rep
15	Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4	1102	Sales Rep
16	Bott	Larry	x2311	lbott@classicmodelcars.com	7	1102	Sales Rep
17	Jones	Bary	x102	bjones@classicmodelcars.com	7	1102	Sales Rep
18	Fitter	Andy	x101	afitter@classicmodelcars.com	6	1088	Sales Rep
19	Marsh	Peter	x102	pmarsh@classicmodelcars.com	6	1088	Sales Rep
20	King	Tom	x103	tking@classicmodelcars.com	6	1088	Sales Rep
21	Nishi	Mani	x101	mnishi@classicmodelcars.com	5	1056	Sales Rep
22	Kato	Yoshimi	x102	ykato@classicmodelcars.com	5	1621	Sales Rep
23	Gerard	Martin	x2312	mgerard@classicmodelcars.com	4	1102	Sales Rep

Query executed successfully. DESKTOP-041884\SQLEXPRESS ... DESKTOP-041884\David ... classicmodels 00:00:00 23 rows

Rysunek 11 Wyświetlenie zawartości tabeli "employees"

SQLQuery1.sql - DESKTOP-041884\SQLEXPRESS:classicmodels (DESKTOP-041884\David - sieci (52)) - Microsoft SQL Server Management Studio

Object Explorer: classicmodels

Query: select * from offices;

officeCode	city	phone	addressLine1	addressLine2	state	country	postalCode	territory
1	San Francisco	+1 550 219 4782	100 Market Street	Suite 300	CA	USA	94080	NA
2	Boston	+1 215 937 0825	1550 Court Place	Suite 102	MA	USA	02107	NA
3	NYC	+1 212 555 3000	523 East 53rd Street	apt. 5A	NY	USA	10022	NA
4	Paris	+33 14 723 4404	43 Rue Jouffroy D'abbans	NULL	NULL	France	75017	EMEA
5	Tokyo	+81 33 224 5000	4-1 Koicho	NULL	Chiyoda-Ku	Japan	102-8578	Japan
6	Sydney	+61 2 9264 2451	5-11 Wentworth Avenue	Floor #2	NULL	Australia	NSW 2010	APAC
7	London	+44 20 7877 2041	25 Old Broad Street	Level 7	NULL	UK	EC2N 1HN	EMEA

Query executed successfully. DESKTOP-041884\SQLEXPRESS ... DESKTOP-041884\David ... classicmodels 00:00:00 7 rows

Rysunek 12 Wyświetlenie zawartości tabeli "offices"

2.4 Import danych

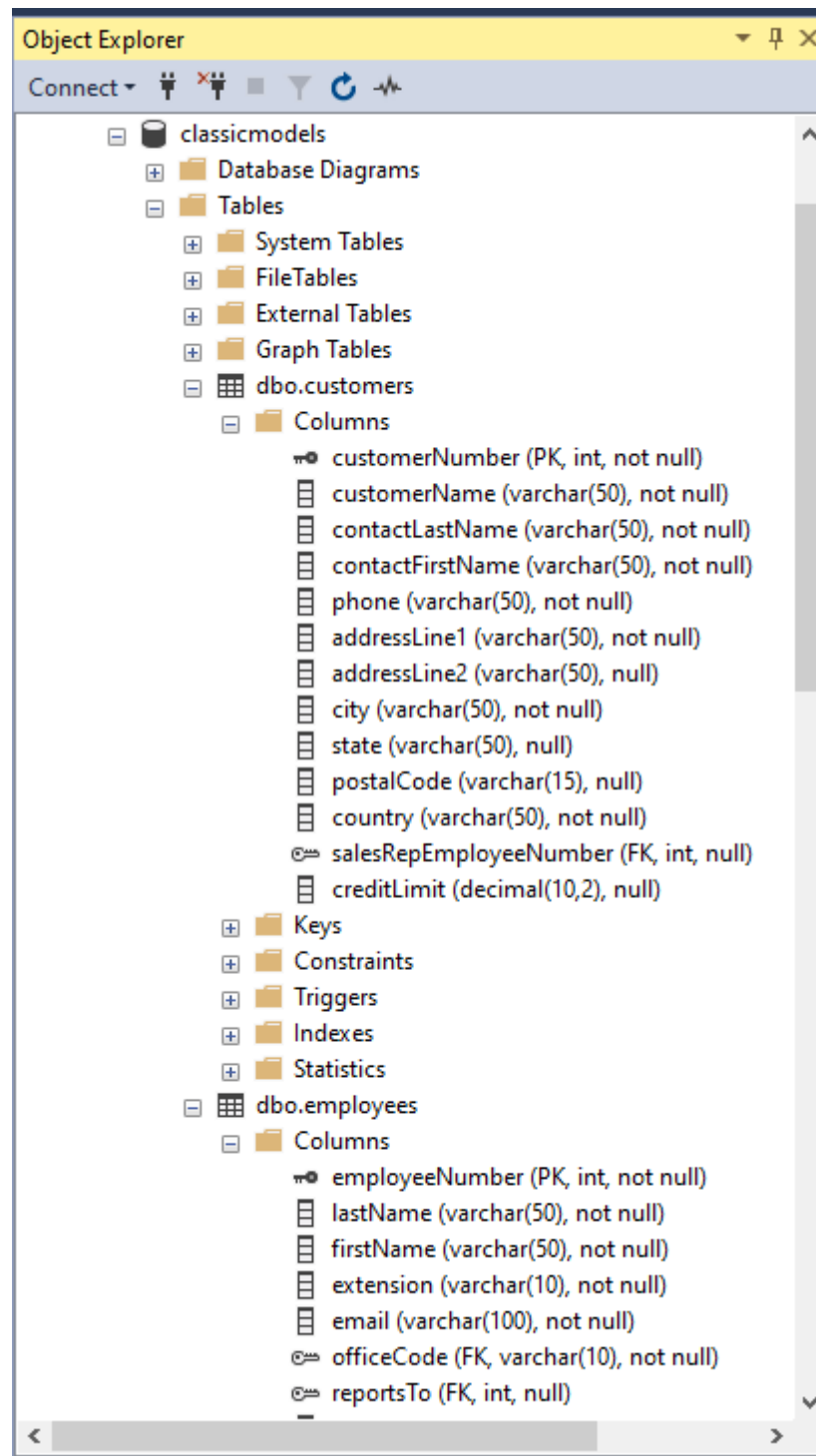
Poniżej zaprezentowany jest fragment kody bazy „classicmodels”, który zostanie zaimportowany do dwóch maszyn z przygotowanymi wcześniej środowiskami.

```
7  /* Create the database */
8  CREATE DATABASE classicmodels;
9
10 /* Switch to the classicmodels database */
11 USE classicmodels;
12
13 /* Drop existing tables */
14 DROP TABLE IF EXISTS productlines;
15 DROP TABLE IF EXISTS products;
16 DROP TABLE IF EXISTS offices;
17 DROP TABLE IF EXISTS employees;
18 DROP TABLE IF EXISTS customers;
19 DROP TABLE IF EXISTS payments;
20 DROP TABLE IF EXISTS orders;
21 DROP TABLE IF EXISTS orderdetails;
22
23 /* Create the tables */
24 CREATE TABLE productlines (
25   productline varchar(50),
26   textDescription varchar(4000) DEFAULT NULL,
27   PRIMARY KEY (productline)
28 );
29
30 CREATE TABLE products (
31   productCode varchar(15),
32   productName varchar(70) NOT NULL,
33   productline varchar(50) NOT NULL,
34   productScale varchar(10) NOT NULL,
35   productVendor varchar(50) NOT NULL,
36   productDescription text NOT NULL,
37   quantityInStock int NOT NULL,
38   buyPrice decimal(10,2) NOT NULL,
39   MSRP decimal(10,2) NOT NULL,
40   PRIMARY KEY (productCode),
41   FOREIGN KEY (productline) REFERENCES productlines (productline)
42 );
43
44 CREATE TABLE offices (
45   officeCode varchar(10),
46   city varchar(50) NOT NULL,
47   phone varchar(50) NOT NULL,
48   addressline1 varchar(50) NOT NULL,
49   addressline2 varchar(50) DEFAULT NULL,
50   state varchar(50) DEFAULT NULL,
51   country varchar(50) NOT NULL,
52   postalCode varchar(15) NOT NULL,
53   territory varchar(10) NOT NULL,
54   PRIMARY KEY (officeCode)
55 );
56
57 CREATE TABLE employees (
58   employeeNumber int,
59   lastName varchar(50) NOT NULL,
60   firstName varchar(50) NOT NULL,
61   extension varchar(10) NOT NULL,
62   email varchar(100) NOT NULL
```

Rysunek 13 Fragment kodu SQL pobranej bazy danych

2.4.1 Dla SSMS

Do zaimportowania bazy danych „classicmodels” wykorzystałem wygenerowany kod SQL z pobranej bazy danych, a następnie go zmodyfikowałem, ponieważ SSMS nie rozpoznawał typu danych t.j „blob” czy „smallint”.



Rysunek 14 Fragment typów zaimportowanych danych

Podczas importu danych do stworzonych wcześniej tabel również wystąpiły problemy np.

```
insert into products(productCode,productName,productline,productScale,productVendor,productDescription,quantityInStock,buyPrice,MSRP) values
('S10_1678','1969 Harley Davidson Ultimate Chopper','Motorcycles','1:10','Min Lin Diecast','This replica features working kickstand, front suspension, gear-shift lever, footbrake l

insert into offices(officeCode,city,phone,addressLine1,addressLine2,state,country,postalCode,territory) values
('1','San Francisco','+1 650 219 4782','100 Market Street','Suite 300','CA','USA','94080','NA'),('2','Boston','+1 215 837 0825','1550 Court Place','Suite 102','MA','USA','02107','W

insert into employees(employeeNumber,lastName,firstName,extension,email,officeCode,reportsTo,jobTitle) values
(1002,'Murphy','Diane','x5800','dmurphy@classicmodelcars.com','1',NULL,'President'),(1056,'Patterson','Mary','x4611','mpatterson@classicmodelcars.com','1',1002,'VP Sales'),(1076,'Fi

insert into customers(customerNumber,customerName,contactLastName,contactFirstName,phone,addressLine1,addressLine2,city,state,postalCode,country,salesRepEmployeeNumber,creditLimit)
(103,'Atelier graphique','Schmitt','Carine ','40.32.2555','54, rue Royale',NULL,'Nantes',NULL,'44000','France',1370,'21000.00'),(112,'Signal Gift Stores','King','Jean','7025551838'

insert into orders(orderNumber,orderDate,requiredDate,shippedDate,status,comments,customerNumber) values
(10100,'2003-01-06','2003-01-13','2003-01-10','Shipped',NULL,363),(10101,'2003-01-09','2003-01-18','2003-01-11','Shipped','Check on availability.',128),(10102,'2003-01-10','2003-01-
```

100 % 4

Messages

Msg 102, Level 15, State 1, Line 128
Incorrect syntax near 's'.

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

Msg 319, Level 15, State 1, Line 128
Incorrect syntax near the keyword 'with'. If this statement is a common table expression, an xmlnamespaces clause or a change tracking context clause, the previous statement must be

100 % 4

Query completed with errors. DESKTOP-0141884\SQLEXPRESS ... DESKTOP-0141884\Dawid ... classicmodels_2 00:00:00 0 rows

Rysunek 15 Problem przy poleceniu "insert"

Problem polegał na tym, że w tekście występował apostrof, który był traktowany jako koniec tekstu:

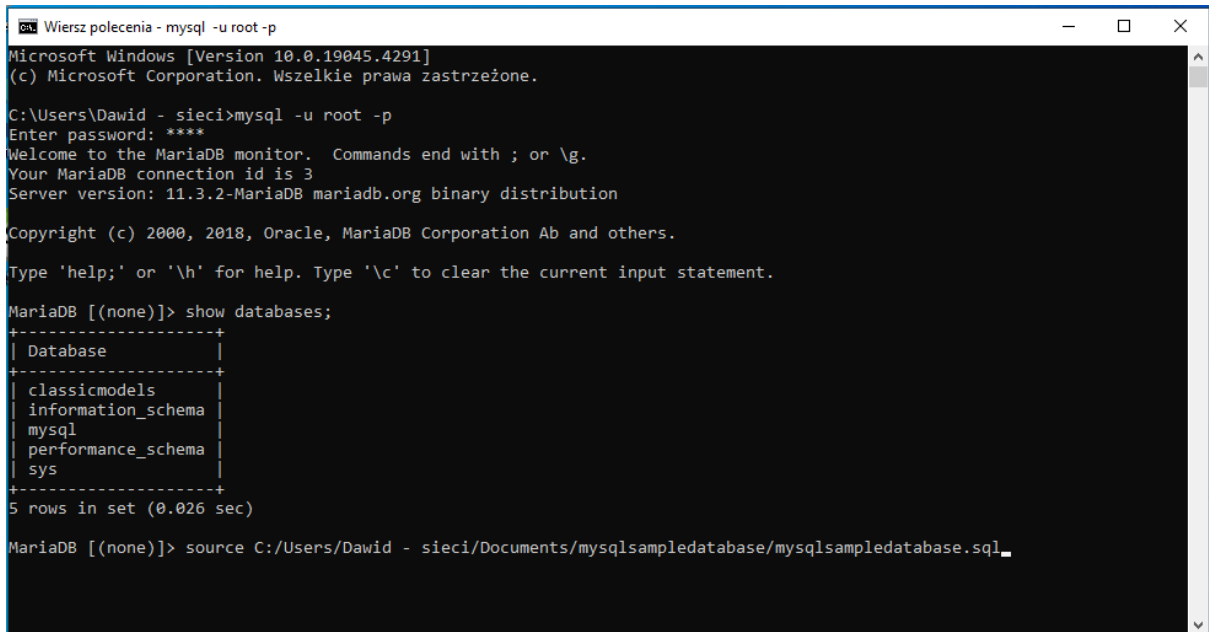
```
ast','1:18 scale replica of actual Army\'s UH-60L BLACK HAWK Helicopter. 100% hand-as
```

Rysunek 16 Znalezienie źródła problemu

Należało usunąć zbędne apostrofy i wszystko pomyślnie zostało zaimportowane.

2.4.2 Dla MariaDB

W przypadku drugiej maszyny wystarczyło użyć polecenia `source` i wskazać ścieżkę do pliku, w którym znajdował się plik z bazą – w naszym przypadku było to:



```
Wiersz polecenia - mysql -u root -p
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Dawid - sieci>mysql -u root -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.3.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| classicmodels |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.026 sec)

MariaDB [(none)]> source C:/Users/Dawid - sieci/Documents/mysqlsampledatabse/mysqlsampledatabse.sql_
```

Rysunek 17 Import bazy w środowisku MariaDB w wierszu poleceń

2.5 Wnioski z procesu importowania bazy danych

2.5.1 SSMS

Import danych w tym systemie był długi, problematyczny oraz niekomfortowy do codziennej pracy, ponieważ nie udało mi się znaleźć szybkiego rozwiązania istniejących problemów jak tylko ręcznie poprawiać błędy, co było bardzo czasochłonne. Nie oznacza to, że system ten jest zły, bo gdy mówimy o prezentacji zaimportowanych danych to SSMS idealnie się sprawdza, możemy czytelnie i przejrzysto odczytać co znajduje się w każdej tabeli oraz bardzo komfortowe jest okno nawigatora z lewej strony, w którym możemy podglądać strukturę tabel oraz ich typy, klucze czy też relacje.

2.5.2 MariaDB

Importowanie danych w środowisku MariaDB również miało swoje wyzwania, ale ogólnie proces ten był bardziej elastyczny i mniej czasochłonny niż w SSMS. Dzięki narzędziom takim jak phpMyAdmin czy HeidiSQL, możliwe było łatwiejsze zarządzanie błędami i szybkie ich rozwiązywanie. MariaDB dostarczała bardziej intuicyjne narzędzia do importu, które często automatyzowały wiele procesów.

Podczas importu za pomocą phpMyAdmin, interfejs użytkownika okazał się być bardzo przyjazny, a możliwości importu z różnych formatów plików (np. CSV, SQL) były dobrze wspierane. W przypadku napotkania błędów, system dostarczał szczegółowe komunikaty, które umożliwiały szybką diagnozę i poprawę problemów.

Z kolei HeidiSQL oferowała bogaty zestaw narzędzi, który ułatwiał importowanie dużych zbiorów danych. Funkcja podglądu i edycji danych na bieżąco była bardzo pomocna, podobnie jak możliwość łatwego definiowania kluczy głównych i obcych bez konieczności pisania złożonych zapytań SQL.

3. Porównanie pracy obu serwerów podczas ich obciążania

3.1 Polecenia jakim baza danych była obciążana

Poleceniem jakim została obciążona nasza baza danych była to rozbudowana kwerenda wybierająca dane z tabeli takie jak:

- CustomerName – kolumna z tabeli
- FirstName – kolumna z tabeli
- LastName – kolumna z tabeli
- TotalOrders – zagnieżdżona kwerenda zliczająca wszystkie zamówienia z tabeli orders dla których customersName było takie samo jak customersName z tabeli cusotmers
- RecentPayments - zagnieżdżona kwerenda sumująca kwoty dla danego klienta zamówień opłaconych po dniu 01.01.2002
- TotalItemOrdered – funkcja sumująca wszystkie ilości zamówień dla danego klienta
- AveragePayment – funkcja licząca średnią kwotę za zamówienie dla danego klienta
- HighestPriceProduct – funkcja wybierająca najdroższy zakupiony przez zamawiającego produkt

Kwerenda dodatkowo zawiera kilka połączeń tabel i warunek wybierający tylko tych klientów, którzy dokonali płatności pomiędzy dniem 01.01.2003, a 12.31.2003. Dane wyświetlane przez nią są grupowane poprzez CusotmerNumber, CustomerName, FirstName oraz LastName. Są one uporządkowane malejąco względem łącznych kwot zamówień.

Polecenie to zostało przedstawione na poniższych screenach zapisane już w konkretnych środowiskach bazodanowych

SQLQuery2.sql - DESKTOP-041884\SQLEXPRESS\classicmodels (DESKTOP-041884\David - sieci (57)) - Microsoft SQL Server Management Studio

```

SELECT
  c.customerName,
  e.firstName,
  e.lastName,
  (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders,
  (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber AND paymentDate > '2003-01-01') AS recentPayments,
  SUM(od.quantityOrdered) AS totalItemsOrdered,
  AVG(p.amount) AS averagePayment,
  MAX(pr.buyPrice) AS highestPriceProduct
FROM customers c
JOIN payments p ON c.customerNumber = p.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
JOIN products pr ON od.productCode = pr.productCode
WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31')
GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName
ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC;
SELECT
  c.customerName,
  e.firstName,
  e.lastName,
  (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders,
  (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber AND paymentDate > '2003-01-01') AS recentPayments,
  SUM(od.quantityOrdered) AS totalItemsOrdered,
  AVG(p.amount) AS averagePayment,
  MAX(pr.buyPrice) AS highestPriceProduct
FROM customers c
JOIN payments p ON c.customerNumber = p.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
JOIN products pr ON od.productCode = pr.productCode
WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31')
GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName
ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC;

```

customerName	firstName	lastName	totalOrders	recentPayments	totalItemsOrdered	averagePayment	highestPriceProduct
1 Euro+ Shopping Channel	Gerard	Hernandez	26	715738.98	121251	55056.844615	101.51
2 Mini Gifts Distributors Ltd.	Leslie	Jennings	17	584188.24	57294	64909.804444	103.42
3 Australian Collectors Co.	Andy	Fixter	5	180585.07	7704	45146.267500	103.42
4 Muscle Machine Inc.	Foon Yue	Tseng	4	177913.95	7100	44478.487500	101.51
5 Dragon Souvenirs, Ltd.	Mami	Nishi	5	156251.03	6096	39062.757500	103.42
6 Down Under Souvenirs, Inc.	Peter	Marsh	5	154622.08	6764	38655.520000	101.51
7 AV Stores, Co.	Larry	Bott	3	148410.09	5334	49470.030000	95.59
8 Anna's Decorations, Ltd.	Andy	Fixter	4	137034.22	5876	34258.555000	103.42

Rysunek 18 Kwerenda zastosowana w SSMS

Unnamed\classicmodels - HeidiSQL 12.6.0.6765

```

SELECT
  c.customerName,
  e.firstName,
  e.lastName,
  (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders,
  (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber AND paymentDate > '2003-01-01') AS recentPayments,
  SUM(od.quantityOrdered) AS totalItemsOrdered,
  AVG(p.amount) AS averagePayment,
  MAX(pr.buyPrice) AS highestPriceProduct
FROM customers c
JOIN payments p ON c.customerNumber = p.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
JOIN products pr ON od.productCode = pr.productCode
WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31')
GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName
ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC;
SELECT
  c.customerName,
  e.firstName,
  e.lastName,
  (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders,
  (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber AND paymentDate > '2003-01-01') AS recentPayments,
  SUM(od.quantityOrdered) AS totalItemsOrdered,
  AVG(p.amount) AS averagePayment,
  MAX(pr.buyPrice) AS highestPriceProduct
FROM customers c
JOIN payments p ON c.customerNumber = p.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
JOIN products pr ON od.productCode = pr.productCode
WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31')
GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName
ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC;

```

customerName	firstName	lastName	totalOrders	recentPayments	totalItemsOrdered	averagePayment	highestPriceProduct
1 Euro+ Shopping Channel	Gerard	Hernandez	26	715 738,98	121 251	55 056,844615	101,51
2 Mini Gifts Distributors Ltd.	Leslie	Jennings	17	584 188,24	57 294	64 909,804444	103,42
3 Australian Collectors Co.	Andy	Fixter	5	180 585,07	7 704	45 146,2675	103,42
4 Muscle Machine Inc.	Foon Yue	Tseng	4	177 913,95	7 100	44 478,4875	101,51
5 Dragon Souvenirs, Ltd.	Mami	Nishi	5	156 251,03	6 096	39 062,7575	103,42
6 Down Under Souvenirs, Inc.	Peter	Marsh	5	154 622,08	6 764	38 655,52	101,51
7 AV Stores, Co.	Larry	Bott	3	148 410,09	5 334	49 470,03	95,59
8 Anna's Decorations, Ltd.	Andy	Fixter	4	137 034,22	5 876	34 258,555	103,42
9 Corporate Gift Ideas Co.	Leslie	Jennings	4	132 340,78	2 894	66 170,39	103,42
10 Saveley & Henriot, Co.	Lou	Bondur	3	130 305,35	4 284	43 435,116667	103,42
11 Rovell Gifts	Pamela	Castillo	3	127 529,69	4 950	42 509,896667	86,7
12 Reims Collectables	Lou	Bondur	5	126 983,19	5 732	31 745,7975	95,59
13 Online Diecast Creations Co.	Steve	Patterson	3	116 449,29	3 744	38 816,43	103,42
14 Kelly's Gift Shop	Peter	Marsh	4	114 497,19	4 941	38 165,73	91,02
15 Corrida Auto Replicas, Ltd	Martin	Gerard	3	112 440,09	3 489	37 480,03	103,42
16 Land of Toys Inc.	George	Vanau	4	107 639,94	4 893	35 879,98	101,51

Rysunek 19 Kwerenda zastosowana w HeidiSQL – środowisko MariaDB

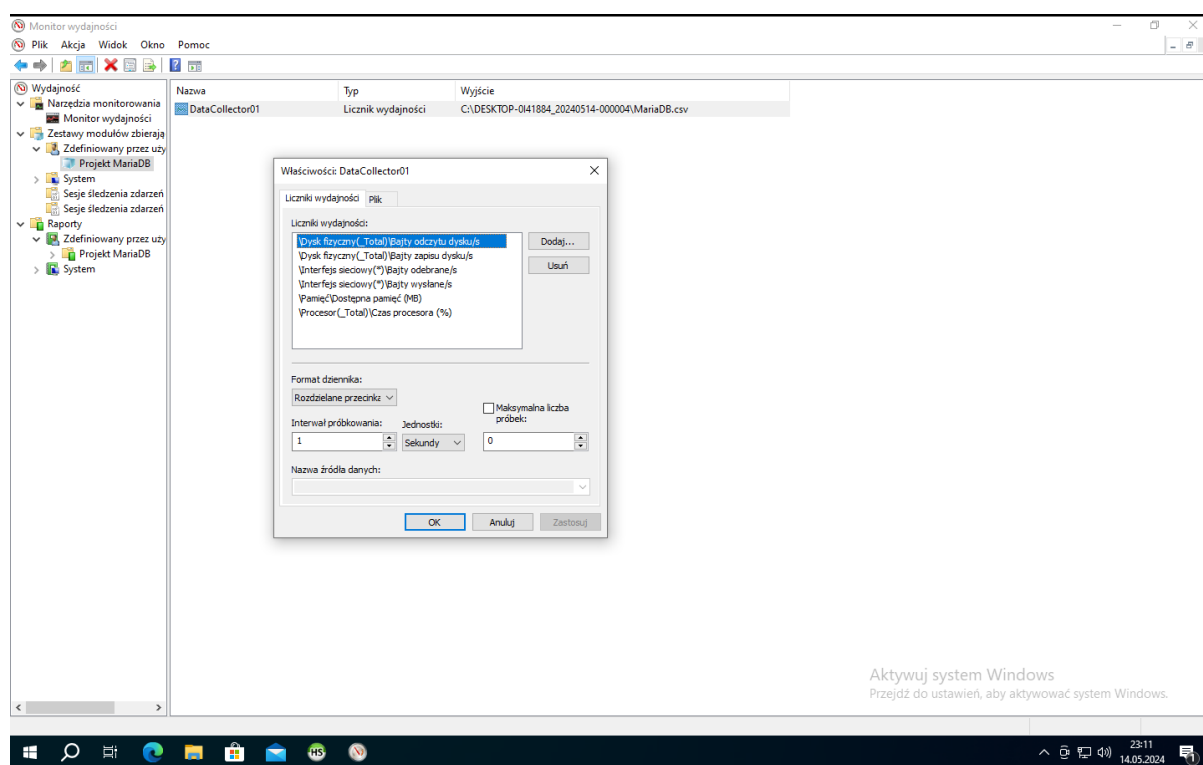
3.2 Porównanie wyników podczas obciążania serwera

3.2.1 Sposób zbierania danych na temat obciążenia

Dane na temat obciążeń kolejnych podzespołów w systemie zostały zbierane za pomocą skonfigurowanego monitora wydajności. Monitor ten co sekunde zbierał dane dotyczące takiej informacji jak:

- Bajty odczytu dysku fizycznego / s
- Bajty zapisu dysku fizycznego / s
- Bajty odebrane przez interfejs sieciowy / s
- Bajty wysłane przez interfejs sieciowy / s
- Czas pracy procesora %

Poniżej pokazany jest skonfigurowany monitor wydajności dla obu maszyn wirtualnych, na których zainstalowano porównywane przez nas serwery bazodanowe



Rysunek 20 Skonfigurowany monitor wydajności

Podczas obciążania serwera polecenie zostało zapisane 2795 razy aby wywołać ciągłą pracę bazy podczas przeszukiwania danych i jak najbardziej obciążyć podzespoły aby można było sprawdzić, która aplikacja bazodanowa lepiej sobie radzi podczas pracy przy bardzo wymagających warunkach.

3.2.2 Porównanie wyników wydajnościowych

Zaczynając nasze porównanie największym parametrem, który nas zaskoczył podczas testów był czas wykonywania polecenia. Na serwerze MariaDB wynosił on nie całe 2 minuty co widać na screenie a dla drugiego serwera czas ten przekroczył ponad 30 minut. Testy były wykonywane przy takich samych warunkach pracy systemu co jeszcze bardziej zaskakuje taka wielka różnica pomiędzy oba serwerami

The screenshot shows the HeidiSQL interface with a complex SQL query and its results. The query is a complex JOIN involving customers, employees, orders, and payments tables, filtered by a date range. The results table shows columns for customer details, order counts, recent payments, total items ordered, average payment, and highest price product. The status bar at the bottom indicates the query was executed successfully on MariaDB 11.3.2.

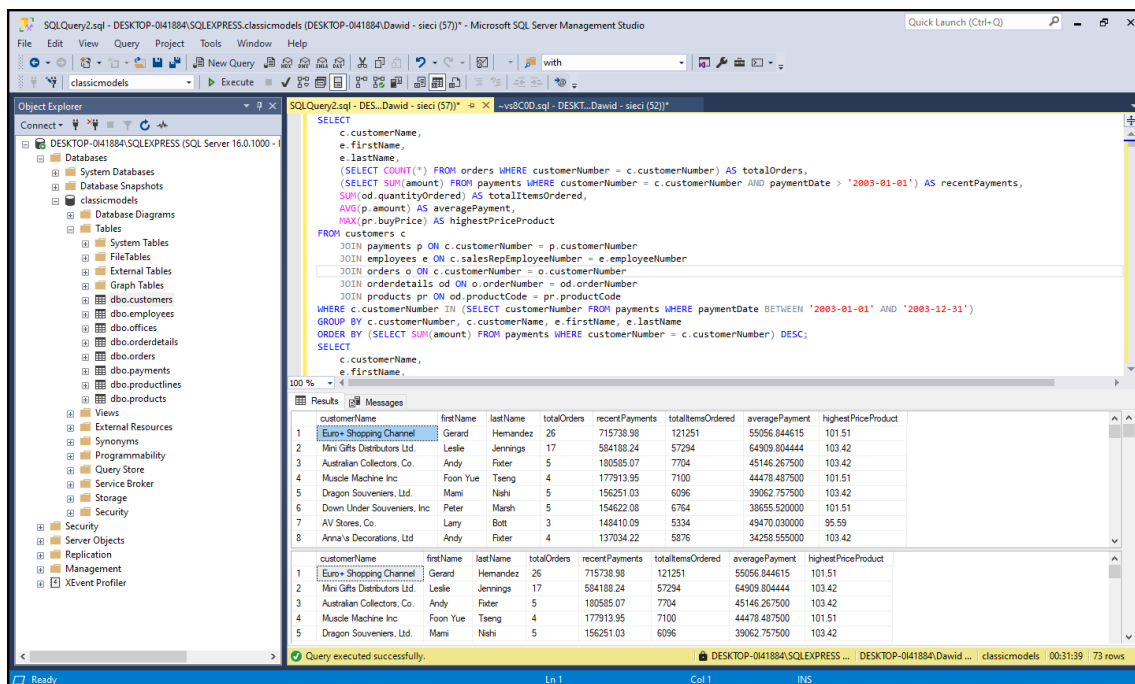
```
1 SELECT
2   c.customerName,
3   e.firstName,
4   e.lastName,
5   (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders,
6   (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber AND paymentDate > '2003-01-01') AS recentPayment,
7   SUM(od.quantityOrdered) AS totalItemsOrdered,
8   AVG(p.amount) AS averagePayment,
9   MAX(pr.buyPrice) AS highestPriceProduct
10  FROM customers c
11  JOIN payments p ON c.customerNumber = p.customerNumber
12  JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
13  JOIN orders o ON c.customerNumber = o.customerNumber
14  JOIN orderdetails od ON o.orderNumber = od.orderNumber
15  JOIN products pr ON od.productCode = pr.productCode
16  WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31')
17  GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName
18  ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC;
19  SELECT
```

#	customerName	firstName	lastName	totalOrders	recentPayments	totalItemsOrdered	averagePayment	highestPriceProduct
1	Euro+ Shopping Channel	Gerard	Hernandez	26	715 738,98	121 251	55 056,844615	101,51
2	Mini Gifts Distributors Ltd.	Leslie	Jennings	17	584 188,24	57 294	64 909,804444	103,42
3	Australian Collectors, Co.	Andy	Fixter	5	180 585,07	7 704	45 146,2675	103,42
4	Muscle Machine Inc.	Foon Yue	Tseng	4	177 913,95	7 100	44 478,4875	101,51
5	Dragon Souvenirs, Inc.	Mami	Nishi	5	156 251,03	6 096	39 062,7575	103,42
6	Down Under Souvenirs, Inc.	Peter	Marsh	5	154 622,08	6 764	38 655,52	101,51
7	AV Stores, Co.	Larry	Bott	3	148 410,09	5 334	49 470,03	95,59
8	Anna's Decorations, Ltd	Andy	Fixter	4	137 034,22	5 876	34 258,555	103,42
9	Corporate Gift Ideas Co.	Leslie	Jennings	4	132 340,78	2 894	66 170,39	103,42
10	Saveley & Henriot, Co.	Loui	Bondur	3	130 305,35	4 284	43 435,116667	103,42
11	Rovell Gifts	Pamela	Castillo	3	127 529,69	4 950	42 509,896667	86,7
12	Reims Collectables	Loui	Bondur	5	126 983,19	5 732	31 745,7975	95,59
13	Online Decat Creations Co.	Steve	Patterson	3	116 449,29	3 744	38 816,43	103,42
14	Kelly's Gift Shop	Peter	Marsh	4	114 497,19	4 941	38 165,73	91,02
15	Corrida Auto Replicas, Ltd	Martin	Gerard	3	112 440,09	3 489	37 480,03	103,42
16	Land of Toys Inc.	George	Vanauf	4	107 639,94	4 893	35 879,98	101,51

2917 SELECT c.customerName, e.firstName, e.lastName, (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders, (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) AS recentPayment, SUM(od.quantityOrdered) AS totalItemsOrdered, AVG(p.amount) AS averagePayment, MAX(pr.buyPrice) AS highestPriceProduct FROM customers c JOIN payments p ON c.customerNumber = p.customerNumber JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber JOIN orders o ON c.customerNumber = o.customerNumber JOIN orderdetails od ON o.orderNumber = od.orderNumber JOIN products pr ON od.productCode = pr.productCode WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31') GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC; 2918 SELECT 2919 SELECT c.customerName, e.firstName, e.lastName, (SELECT COUNT(*) FROM orders WHERE customerNumber = c.customerNumber) AS totalOrders, (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) AS recentPayment, SUM(od.quantityOrdered) AS totalItemsOrdered, AVG(p.amount) AS averagePayment, MAX(pr.buyPrice) AS highestPriceProduct FROM customers c JOIN payments p ON c.customerNumber = p.customerNumber JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber JOIN orders o ON c.customerNumber = o.customerNumber JOIN orderdetails od ON o.orderNumber = od.orderNumber JOIN products pr ON od.productCode = pr.productCode WHERE c.customerNumber IN (SELECT customerNumber FROM payments WHERE paymentDate BETWEEN '2003-01-01' AND '2003-12-31') GROUP BY c.customerNumber, c.customerName, e.firstName, e.lastName ORDER BY (SELECT SUM(amount) FROM payments WHERE customerNumber = c.customerNumber) DESC; 2921 /* Zmodyfikowane wiersze: 0 Znalezione wiersze: 204 984 Ostrzeżenia: 0 Czas przetwarzania: 2 808 zapytania: 00:01:44.1 (+ 0,016 sek. sieć) */

r50 293 : c8 (2,7 MB) Połączony: 13:44 h MariaDB 11.3.2 Czas pracy: 14:13 h Czas serwera: 15:31 Bezczynny.

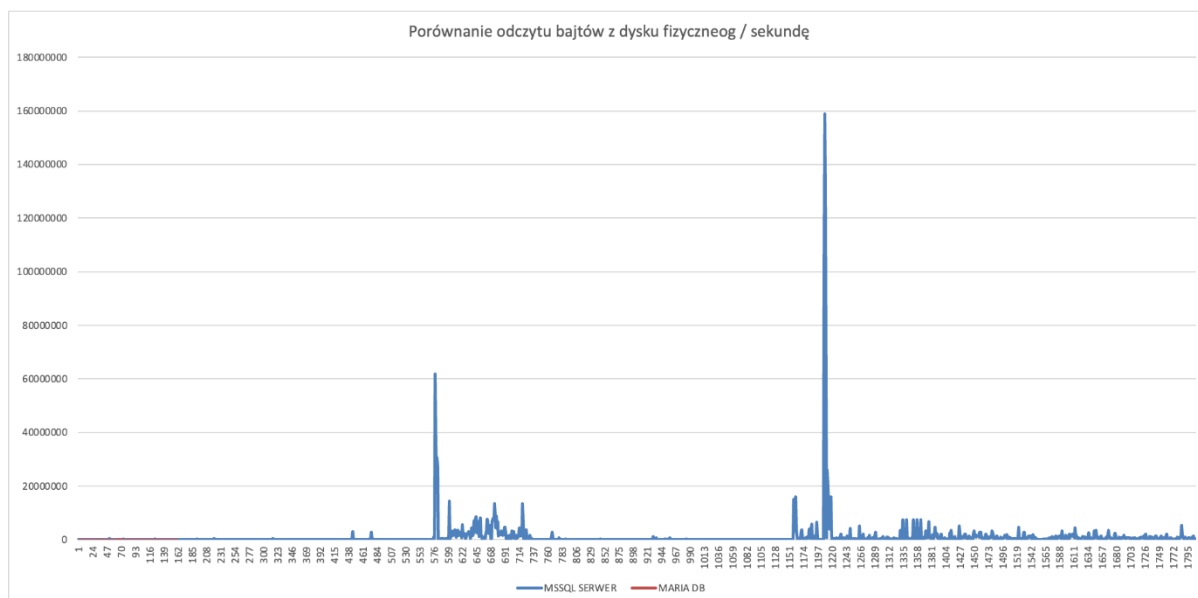
Rysunek 21 Kwerenda nr 2 w HeidiSQL



Rysunek 22 Kwerenda nr 2 w SSMS

Kolejnymi parametrami jakie porównywaliśmy były:

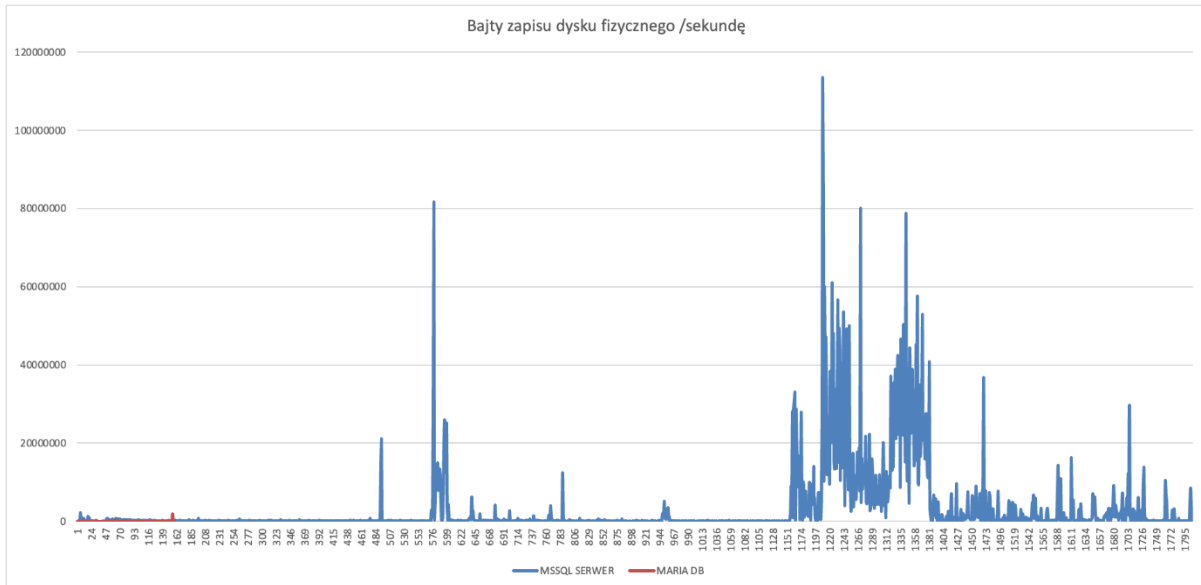
- Bajty odczytu dysku fizycznego



Wykres 1 Wykres przedstawiający bajty odczytu dysku fizycznego

Widać dużo większe obciążenie dysku podczas wykonywania polecenia na MSSQL SERVER oraz widać 2 większe obciążenia w około 10 i 20 minucie.

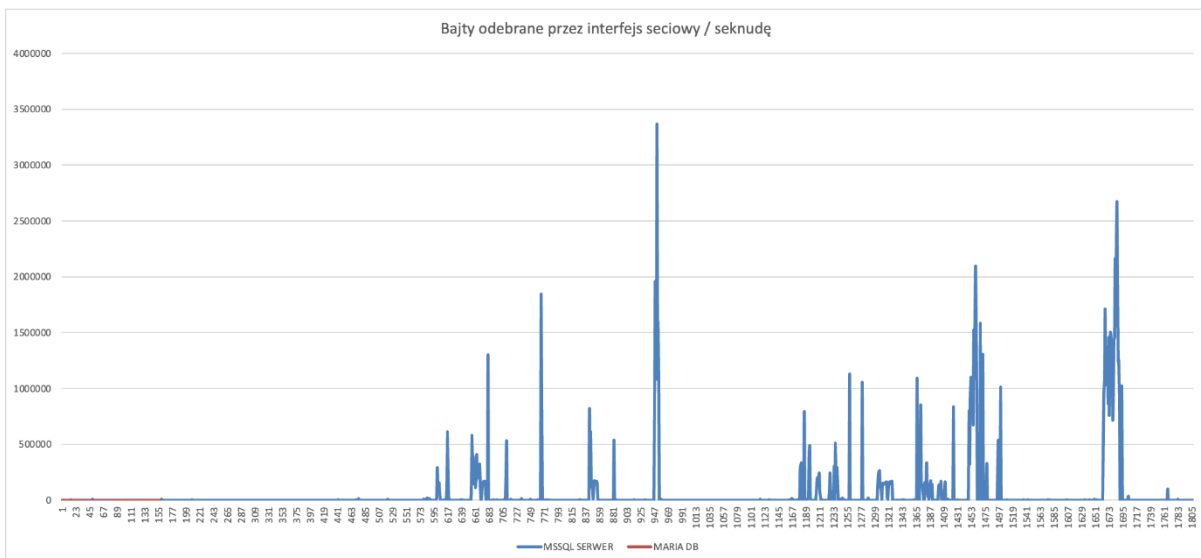
- Bajty zapisu dysku fizycznego



Wykres 2 Wykres przedstawiający bajty zapisu dysku fizycznego

Możemy zaobserwować jeden wzrost wartości pod koniec wykonywania się polecenia na Maria DB oraz kilka małych i krótkich skoków i jeden duży i długi trwający około 3 minuty.

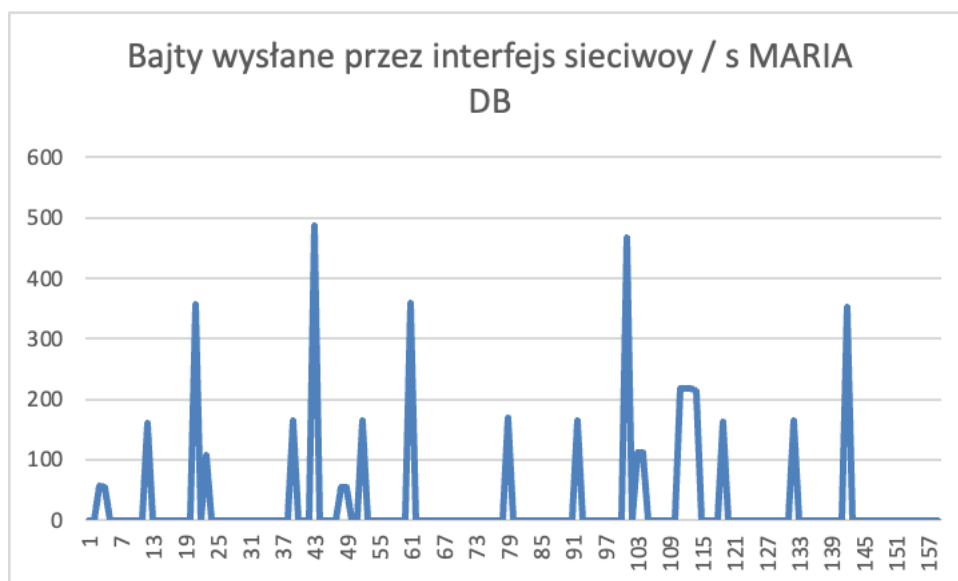
- Bajty odebrane przez interfejs sieciowy/ s



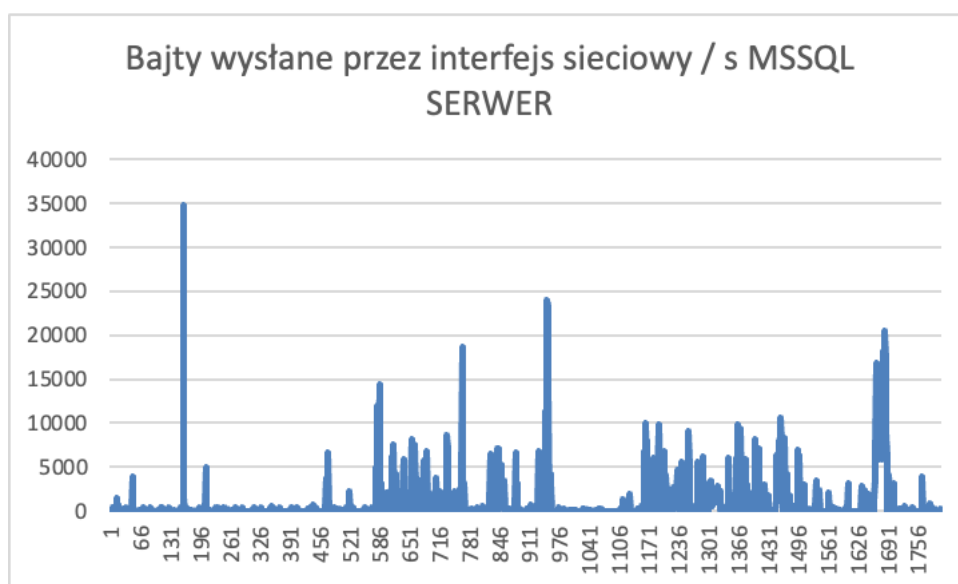
Wykres 3 -Wykres przedstawiający bajty odebrane przez interfejs sieciowy/ s

Możemy zaobserwować jednostajny ruch odbierany podczas pracy Maria DB i bardzo wzrastający do wysokiej wartości i wracający do 0 w przypadku MS SQL SERVER.

- Bajty wysłane przez interfejs sieciowy /s



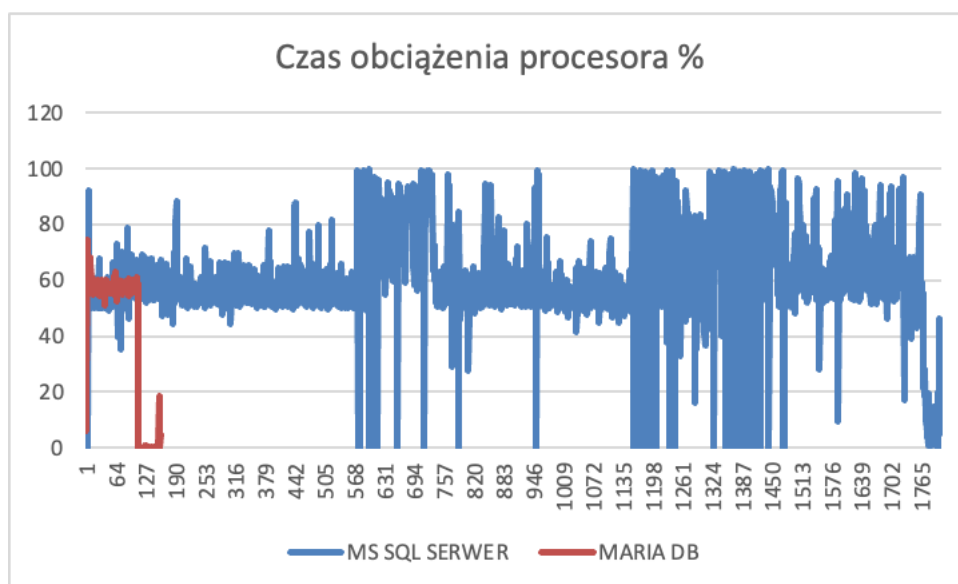
Wykres 4 Wykres przedstawiający bajty wysłane przez interfejs sieciowy /s w środowisku MariaDB



Wykres 5 Wykres przedstawiający bajty wysłane przez interfejs sieciowy /s w środowisku SSMS

Na poniższych wykresach możemy zauważyć chwilowe wzrosty podczas wysyłania danych przez interfejs sieciowy jednak skala ilości bajtów podczas MS SQL SERVER jest dużo większa niż MARIA DB.

- Czas obciążenia procesora %



Wykres 6 Wykres przedstawiający czas obciążenia procesora (%)

Na tym wykresie mamy przedstawiony czas obciążenia procesora. Jak widać najwyższa wartość dla MARIA DB oscyluje w granicach 75% a średnia wartość waha się około 50-60% więc procesor podczas tej pracy nie był bardzo obciążony. W przypadku MSSQL SERWER przez pierwsze 10 minut wartości miały trochę większą amplitudę lecz też wartości znajdowały się w granicach 50-70% dopiero od około 9 minuty wartości te zaczęły skakać od 0 do 100% zużycia procesora.

3.2.3 Podsumowanie

Podsumowując tą analizę na temat pracy działania serwerów podczas obciążania ich kwerendą wybierającą dużo lepiej wypadł serwer MariaDB. Czas pracy w jakim wykonało się zapytanie był 15 razy szybszy niż na MSSQL SERWER. Kolejnym aspektem jest obciążenie podzespołów komputerowych, na których postawiona jest nasza maszyna jest również z dużą korzyścią dla MariaDB. Nie są one używane do granic swoich możliwości dzięki czemu będą mogły dłużej bezawaryjnie cieszyć nas swoją pracą.

4. Bibliografia

- [1] Oficjalna strona MySQL MariaDB - <https://mariadb.org/>
- [2] Oficjalna strona SSMS od firmy Microsoft - <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>
- [3] Dyer, Russell JT. *Learning MySQL and MariaDB: Heading in the right direction with MySQL and MariaDB.* " O'Reilly Media, Inc.", 2015.
- [4] Shears, Stephen. "Data Processing in a Database Management System Using Parallel Processing." (2022).
- [5] DuBois, P. (2013). *MySQL*. Addison-Wesley.

5. Spis rysunków

Rysunek 1 Środowisko bazodanowe MySQL	1
Rysunek 2 Zrzut ekranu skonfigurowanego serwera MySQL.....	3
Rysunek 3 Zrzut ekranu konsoli MySQL - MariaDB.....	4
Rysunek 4 Diagram ER bazy danych "classicmodels"	6
Rysunek 5 Wyświetlenie zawartości tabeli "customers"	7
Rysunek 6 Wyświetlenie zawartości tabeli "products"	7
Rysunek 7 Wyświetlenie zawartości tabeli "productlines"	8
Rysunek 8 Wyświetlenie zawartości tabeli "orders"	8
Rysunek 9 Wyświetlenie zawartości tabeli "orderdetails"	9
Rysunek 10 Wyświetlenie zawartości tabeli "payments"	9
Rysunek 11 Wyświetlenie zawartości tabeli "employees"	10
Rysunek 12 Wyświetlenie zawartości tabeli "offices"	10
Rysunek 13 Fragment kodu SQL pobranej bazy danych.....	11
Rysunek 14 Fragment typów zaimportowanych danych	12
Rysunek 15 Problem przy poleceniu "insert"	13
Rysunek 16 Znalezienie źródła problemu.....	13
Rysunek 17 Import bazy w środowisku MariaDB w wierszu poleceń	14
Rysunek 18 Kwerenda zastosowana w SSMS	17
Rysunek 19 Kwerenda zastosowana w HeidiSQL – środowisko MariaDB	17
Rysunek 20 Skonfigurowany monitor wydajności	18
Rysunek 21 Kwerenda nr 2 w HeidiSQL	19
Rysunek 22 Kwerenda nr 2 w SSMS.....	20

6. Spis wykresów

Wykres 1 Wykres przedstawiający bajty odczytu dysku fizycznego	20
Wykres 2 Wykres przedstawiający bajty zapisu dysku fizycznego.....	21
Wykres 3 -Wykres przedstawiający bajty odebrane przez interfejs sieciowy/ s	21
Wykres 4 Wykres przedstawiający bajty wysłane przez interfejs sieciowy /s w środowisku MariaDB	22
Wykres 5 Wykres przedstawiający bajty wysłane przez interfejs sieciowy /s w środowisku SSMS	22
Wykres 6 Wykres przedstawiający czas obciążenia procesora (%)	23