Odsłaniamy SQL Server 2019

Klastry Big Data
i uczenie maszynowe

—

Bob Ward
przedmowa Rohan Kumar

Apress®

Apress®

Bob Ward

Odsłaniamy SQL Server 2019

Klastry Big Data i uczenie maszynowe

Przekład: Marek Włodarz

APN Promise, Warszawa 2020

Odsłaniamy SQL Server 2019. Klastry Big Data i uczenie maszynowe

First published in English under the title
SQL Server 2019 Revealed: Including Big Data Clusters and Machine Learning
by Bob Ward, edition: 1

Copyright © 2019 by Bob Ward

This edition has been translated and published under licence from APress Media, LLC, part of Springer Nature.

APress Media, LLC, part of Springer Nature takes no responsibility and shall not be made liable for the accuracy of the translation.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from publisher.

Polish language edition published by APN PROMISE S.A., Copyright © 2020

Autoryzowany przekład z wydania w języku angielskim, zatytułowanego: SQL Server 2019 Revealed: Including Big Data Clusters and Machine Learning, by Bob Ward, edition: 1, opublikowanego przez APress Media, LLC, oddział Springer Nature.

Wszystkie prawa zastrzeżone. Żadna część niniejszej książki nie może być powielana ani rozpowszechniana w jakiejkolwiek formie i w jakikolwiek sposób (elektroniczny, mechaniczny), włącznie z fotokopiowaniem, nagrywaniem na taśmy lub przy użyciu innych systemów bez pisemnej zgody wydawcy.

APN PROMISE SA, ul. Domaniewska 44a, 02-672 Warszawa
tel. +48 22 35 51 600, fax +48 22 35 51 699
e-mail: mspress@promise.pl

Książka ta przedstawia poglądy i opinie autorów. Przykłady firm, produktów, osób i wydarzeń opisane w niniejszej książce są fikcyjne i nie odnoszą się do żadnych konkretnych firm, produktów, osób i wydarzeń, chyba że zostanie jednoznacznie stwierdzone, że jest inaczej. Ewentualne podobieństwo do jakiejkolwiek rzeczywistej firmy, organizacji, produktu, nazwy domeny, adresu poczty elektronicznej, logo, osoby, miejsca lub zdarzenia jest przypadkowe i niezamierzone.

Wszystkie znaki towarowe występujące w książce mogą być własnością ich odnośnych właścicieli.

APN PROMISE SA dołożyła wszelkich starań, aby zapewnić najwyższą jakość tej publikacji. Jednakże nikomu nie udziela się rękojmi ani gwarancji.
APN PROMISE SA nie jest w żadnym wypadku odpowiedzialna za jakiekolwiek szkody będące następstwem korzystania z informacji zawartych w niniejszej publikacji, nawet jeśli APN PROMISE została powiadomiona o możliwości wystąpienia szkód.

ISBN: 978-83-7541-417-2 (druk), 978-83-7541-419-6 (ebook)

Przekład: Marek Włodarz
Korekta: Ewa Swędrowska
Skład i łamanie: MAWart Marek Włodarz

Książkę tę dedykuję społeczności SQL Server Community, znanej też jako #sqlfamily. Bez was ten wspaniały produkt nie byłby tym, czym jest dzisiaj.

Spis treści

O autorze

Bob Ward zajmuje stanowiska naczelnego architekta (Principal Architect) w zespole Microsoft Azure Data SQL Server, który odpowiada za tworzenie wszystkich wydań SQL Server. Bob od 26 lat pracuje w firmie Microsoft, przy każdej wersji SQL Server, poczynając od wydania dla OS/2 1.1, aż po SQL Server 2019, w tym Azure. Jest dobrze znanym wykładowcą na temat SQL Server, często prowadzącym prezentacje na temat nowych wydań, wewnętrznych mechanizmów i wydajności na takich imprezach, jak PASS Summit, SQLBits, SQLIntersection, Red Hat Summit, Microsoft Inspire i Microsoft Ignite. Można go śledzić na Twitterze (@bobwardms) lub na `www.linkedin.com/in/bobwardms`. Jest autorem książki *Pro SQL Server on Linux*, opublikowanej przez Apress Media.

O recenzencie technicznym

Aaron Bertrand jest zaangażowanym technologiem z ponad dwudziestoletnim doświadczeniem w dziedzinie SQL Server. Pracował bezpośrednio w wielu zespołach produktowych w firmie Microsoft i jest dobrze znany z pomagania w podnoszeniu umiejętności technicznych szerokiej społeczności programistów poprzez wypowiedzi i moderowanie na różnych grupach technicznych.

Przedmowa

Znajdujemy się w unikatowym punkcie zwrotnym w historii technologii i nigdy nie było lepszego czasu, aby pracować w obszarze danych, analityki i sztucznej inteligencji. Tempo rozwoju w dziedzinie przetwarzania danych jest szybsze, niż kiedykolwiek wcześniej, zaś przełom cyfrowy wprowadzony przez technologie AI i ML stwarza firmom nieograniczony potencjał wykorzystania danych jako przewagi konkurencyjnej. Wraz z dramatycznym przyśpieszeniem cyfryzacji podstawowym wyzwaniem, przed którym stoimy obecnie, jest to, jak wykorzystać te ogromne ilości danych, aby pomóc w transformacji naszych firm i społeczeństw.

Dostrzegamy wielkie możliwości napędzane przez inteligentną chmurę i inteligentny brzeg. SQL Server jest produktem niezrównanym, jeśli chodzi o poziom spójności zapewniany programistom, inżynierom danych i administratorom we wszystkich obszarach – w instalacji lokalnej, chmurze prywatnej i publicznej. Społeczność SQL Server odgrywała bardzo ważną rolę w jego ewolucji i nie mogę dostatecznie wyrazić tego, jak bardzo jesteśmy wdzięczni za pomoc i informacje zwrotne, jakie otrzymujemy od ponad 25 lat.

SQL Server 2019 jest wydaniem fenomenalnym i jestem dumny z tego, co zespół zdołał dostarczyć. SQL Server 2019 oparty jest na innowacjach, które zostały wprowadzone w wersjach SQL Server 2016 i SQL Server 2017. Podczas gdy jest tu wiele nowych możliwości, które dobrze będą służyć wielu klientom, czego można oczekiwać od każdego głównego wydania SQL Server, najbardziej poruszają mnie te innowacje, które rozszerzają umiejętności budowane przez naszych klientów od dziesięcioleci i pozwolą im zarządzać i uzyskiwać korzyści z systemów Big Data. Ta innowacja odgrywa krytyczną rolę w cyfrowej transformacji.

Bob Ward związany jest z zespołem SQL Server od samego początku rozwoju produktu i miał znaczący wpływ na jego kształt. Niewielu jest takich, którzy tak szeroko i dogłębnie rozumieją to, co on robi. Można to zauważyć w tym, jak radzi sobie z wyjaśnianiem złożonych koncepcji w prosty, łatwy do zrozumienia sposób, tak jak w tej książce. Życzę wszystkim przyjemnej lektury.

Rohan Kumar
Corporate Vice President, Azure Data w Microsoft

Podziękowania

W moim życiu zdarzyło się wiele rzeczy, za które jestem wdzięczny, a jedną z nich jest to, że byłem w stanie napisać tę książkę. Na początek muszę podziękować mojej pięknej i utalentowanej żonie, Ginger. Jest moim partnerem i bratnią duszą. Niestrudzenie słuchała moich narzekać, cierpliwie znosiła długie wieczory spędzane przeze mnie nad książką, a niekiedy musiała być moim kierowcą, dzięki czemu mogłem pracować w fotelu pasażera. Nie znam nikogo, kto miałby większe szczęście. W tym roku obchodziliśmy 30 rocznicę ślubu i nadal kochamy się tak bardzo, jak to możliwe. Chciałbym też podziękować moim synom, Troyowi i Ryanowi. Troy mieszka obecnie w Charleston w Południowej Karolinie, gdzie pisałem ostatnią część tej książki, akurat wtedy, gdy nadchodził huragan Dorian. Troy jest kimś, kogo cenię nie tylko za jego charakter, ale za jego dążenia do uczynienia świata lepszym. Ryan jest na drugim roku studiów w Baylor Law School (dalej, Bears!). Nadal zadziwia mnie inteligencją, uporządkowaniem i umiejętnością trzymania wszystkiego razem, a mimo to potrafi znajdować czas, aby doskonalić grę w golfa. Chcę też podziękować mojej mamie, Annette Gibaud, która jest przykładem tego, że można być dobrym dla wszystkich każdego dnia.

Muszę także wspomnieć o wielu ludziach, którzy przyczynili się do powstania tej książki. Najpierw muszę podziękować Apress Media za kolejną możliwość napisania książki. Jonathan Gennick i Jill Balzano ponownie pomagali mi w całej drodze, od pierwszych zdań aż po zakończenie. Książka ta nie mogłaby w ogóle powstać, a na pewno nie na czas, bez mojego recenzenta technicznego, Aarona Bertranda. Gdy tylko zacząłem myśleć o napisaniu tej książki, Aaron był pierwszym, który przyszedł mi na myśl jako recenzent, a to ze względu na jego ogromną wiedzę i reputację eksperta w społeczności SQL Server. Aaron jest po prostu nadczłowiekiem, jeśli pomyślę, jak szybko opracowywał recenzje każdego rozdziału.

Spośród współpracowników z firmy Microsoft, pierwsze i najważniejsze podziękowania należą się Rohanowi Kumarowi, Gayle Sheppardowi i Asadowi Khan za okazję rozpowszechniania wieści o SQL Server 2019, co zapewniło mi szczegółową wiedzę o produkcie, niezbędną do napisania tej książki. Chcę też osobiście podziękować moim najbliższym kolegom, Buckowi Woody i Annie Hoffman (Thomas). Przemierzyłem pół świata wraz z nimi w roku 2018 i 2019, prowadząc prezentacje na temat SQL Server,

Big Data Clusters i Azure. Sprawili, że stałem się lepszym wykładowcą i nauczycielem. O zespole Microsoft SQL Server Engineering można powiedzieć tylko, że jest wspaniały. Jestem głęboko wdzięczny, że mogę pracować z tak inteligentnymi i profesjonalnymi ludźmi, z których wielu pomogło mi przy różnych szczegółach, które czytelnik znajdzie w tej książce. Listę tę otwierają Slava Oks i Travis Wright, którzy pomogli mi opowiedzieć historię projektów Seattle i Aris, a także byli wielką pomocą w poznawaniu większości nowości tego wydania, szczególnie Big Data Clusters. Conor Cunningham nadal zadziwia mnie swoją wiedzą na temat produktu, jednocześnie skutecznie wymuszając wysoką jakość wydania.

Prawdziwi bohaterowie tej książki to członkowie zespołu inżynierskiego, którzy zbudowali to wydanie i pomagali mi w w różnych częściach książki. Wymienię ich bez żadnego szczególnego uporządkowania: Pedro Lopes, Pam Lahoud, Amit Banerjee, Brian Carrig, Tejas Shah, Vin Yu, Sourabh Agarwal, Mihaela Blendea, Nellie Gustafsson, Abiola Oke, James Rowland-Jones, Scott Konersmann, Stuart Padley, David Kryze, Robert Dorr, Mitchell Sternke, Ross Monster, Madeline MacDonald, Dylan Gray, Joe Sack, Shreya Verma, Jakub Szymaszek, Joachim Hammer, Raghav Kaushik, Parag Paul, Panagiotis Antonopoulos, Michael Nelson, Pranjal Gupta, Jarupat Jisarojito, Weiyun Huang, George Reynya, Umachandar Jayachandran (UC), Sahaj Saini, Mike Habben, Vaqar Pirzada, Rony Chatterjee, Vicky Harp, Alan Yu, Jack Li, Alexey Eksarevskiy, Jay Choe, Argenis Fernandez, Kevin Farlee, Arieh Bibliowicz, Alex Umansky, Matteo Taveggia, Kapil Thacker, Li Zhang oraz Dong Cao.

Książka ta i cała moja praca nie byłyby możliwe bez pomocy partnerskich firm, takich jak HPE, DELL i Red Hat, którzy pozwolili mi opowiedzieć historię SQL Server 2019 swoim klientom. Dziękuję Wendy Harms, Billowi Dunmire, Urs Renggli, Robertowi Sondersowi, Louisowi Imersheinowi i Stephane Bureau (mój człowiek od wideo). Specjalne podziękowania należą się też Davidowi DeWitt za wgląd w historię Polybase, Brendanowi Burnsowie za spostrzeżenia i przedstawienie podstaw wiedzy o Kubernetes, a także Anthony'emu Nocentino za ogromną wiedzę o systemie Linux i kontenerach.

Na koniec dziękuję społeczności SQL Server na całym świecie. Przygotowujemy teraz wydania SQL Server szybciej, niż dawniej, ale nadal odczuwam wasz entuzjazm i docenianie naszych starań za każdym razem, gdy prezentuję kolejną wersję SQL Server.

Wprowadzenie

Podobnie jak moja pierwsza książka *Pro SQL Server on Linux*, ta książka przebyła długą drogę, i to dosłownie. W trakcie roku 2019 podróżowałem więcej, niż w całym wcześniejszym życiu. Oznaczało to, że musiałem być gotowy na pisanie w każdym miejscu i czasie, gdy tylko mogłem. Książka powstawała podczas lotów, w hotelach, pociągach, a nawet w podróżach samochodem, w takich miastach jak Seattle, Londyn, Manchester (UK), Nashville, Las Vegas (wiele razy), San Antonio, Austin, Houston, Orlando, St. Lucia (to były wakacje), Genesee (Kolorado), Charleston, Boston, Dubai, Johannesburg (Południowa Afryka), Greenville (Południowa Karolina) oraz Indianapolis, no i wieczorami w moim domowym biurze w North Richland Hills, Teksas.

Po ukończeniu pierwszej książki sądziłem, że nie będę gotów napisać kolejnej, ale nie mogłem odrzucić okazji do opowiedzenia historii o SQL Server 2019. Ta książka naprawdę odzwierciedla znane powiedzenie „pracować z miłością". Włożyłem serce i duszę w uczenie się, uczenie innych, narzekanie, psucie, dokumentowanie, testowanie i posługiwanie się SQL Server 2019. Ta książka reprezentuje to wszystko i jeszcze więcej.

Książkę tę napisałem dla specjalistów danych i programistów, którzy mają ogólną wiedzę na temat SQL Server, ale potrzebują wyczerpującego spojrzenia na SQL Server 2019. Dlatego książka zawiera mnóstwo przykładów, ilustracji i odsyłaczy do źródeł, aby pomóc w poznawaniu produktu. Książka ta powinna nie tylko zapewnić zrozumienie tego, co znajdziemy w SQL Server 2019, ale może posłużyć również jako referencyjne źródło, do którego można powrócić w każdej chwili.

Choć każdy rozdział jest niezależną całością, zdecydowanie polecam rozpoczęcie od rozdziału 1, jako że przedstawia on historię i podstawy tego wydania. Przedstawiłem w nim również kluczowe możliwości SQL Server 2019 i wyjaśniłem, dlaczego uważam go za konkurencyjny produkt. Następnie można czytać rozdziały w kolejności albo pewne pominąć. Jedno jest pewne, aby wydobyć najwięcej z rozdziału 10 na temat Big Data Clusters, trzeba przeczytać najpierw rozdziały 6, 7, 8 i 9.

Książka jest zasadniczo podzielona na następujące główne części:

- Rozdział 1, przedstawiający historię i ogólną postać wydania SQL Server 2019.

- Rozdziały 2, 3 i 4 zajmują się wydajnością, zabezpieczeniami i dostępnością. Sama zawartość tych rozdziałów wystarczy, aby wzbudzić zainteresowanie wersją SQL Server 2019.
- Rozdział 5 samodzielnie przeznaczony jest dla programistów.
- Rozdziały 6, 7 i 8 poświęcone są tematyce Linuxa, kontenerów i Kubernetes.
- Rozdział 9 przedstawia zagadnienie wirtualizacji danych przy użyciu Polybase.
- Rozdział 10 to wielki rozdział na wielki temat: Big Data Clusters.
- Rozdział 11 zamyka książkę omówieniem rozmaitych nowych funkcji oraz migracji.

Jestem zwolennikiem podejścia „uczenia przez przykład", zatem dołączyłem do niemal każdego rozdziału wiele przykładów (a w niektórych przypadkach wyjaśnienia, jak używać przykładów już istniejących). Wszystkie te przykłady można znaleźć w repozytorium GitHub, używając łącza ze strony książki `www.apress.com/9781484254189` albo bezpośrednio w moim repozytorium GitHub pod adresem `https://github.com/microsoft/bobsql`.

Zalecam także przyjrzenie się bezpłatnym źródłom szkoleniowym zbudowanym przez nasz zespół, dostępnym pod adresem `https://aka.ms/sqlworkshops`. Zawierają one także bezpłatne warsztaty szkoleniowe dla SQL Server 2019!

Przy tej książce poświęciłem sporo czasu na rozmyślanie nad tym, „co mógłby chcieć czytelnik" w odniesieniu do określonego tematu lub przykładu. Mam nadzieję, że będzie to można zauważyć i poczuć podczas lektury. Jeśli ktoś będzie miał jakieś pytania lub problemy, chciałbym o nich usłyszeć. Można do mnie pisać bezpośrednio na adres bobward@microsoft.com.

Bob Ward
North Richland Hills, Teksas
wrzesień 2019

Rozdział 1

Dlaczego SQL Server 2019?

W lipcu 2017 roku, jako członek zespołu inżynierskiego SQL Server, odbyłem jedną z moich regularnych podróży do Redmond w stanie Washington. Na co dzień mieszkam w North Richland Hills, w Teksasie, a nowoczesna technologia pozwala na wykonywanie większości mojej pracy w oddaleniu od reszty zespołu SQL Engineering. Należę jednak do starej szkoły i jestem zdania, że w pewnych sytuacjach nic nie może zastąpić wspólnej pracy z innymi ludźmi. W lipcu 2017 należałem już do zespołu SQL Engineering od ponad roku, zajmując się głównie SQL Server 2016 (przykład mojej pracy związanej z SQL Server 2016 można obejrzeć na stronie `https://channel9.msdn.com/Events/Ignite/2016/BRK3043-TS`).

Do tego czasu byłem członkiem sławnego „zespołu tygrysów", ale w ramach mojej wizyty w 2017 roku zostałem poproszony o przyjęcie nowych zadań, skupiających się głównie na nadchodzącym wydaniu SQL Server 2017. Zawierało ono SQL Server dla systemu Linux, co ostatecznie doprowadziło mnie do napisania pierwszej książki, *Pro SQL Server on Linux (*`www.apress.com/us/book/9781484241271`). W ramach tej wizyty zacząłem zatem spotykać się i rozmawiać z różnymi członkami zespołu o SQL Server 2017 – usprawnieniach wydajnościowych, ogólnym zbiorze nowych funkcji i szczegółami związanymi z SQL Server dla Linuksa oraz kontenerami. Jedną z osób, z którymi rozmawiałem podczas tego tygodnia, był Slava Oks. Slava jest wiodącym menedżerem rozwoju dla SQL Server i jednym z pomysłodawców SQL Server on Linux. Napisał przedmowę do *Pro SQL Server on Linux*, zaś rozdział 1 tej książki przedstawia historię jego zaangażowania w ten projekt. W tamtym czasie Slava lubił wcześniej pojawiać się w biurze; gdy przebywam w Redmond, ja również staram się pracować w „czasie Teksasu" – co oznacza, że również przychodziłem bardzo wcześnie. Tym samym często spotykaliśmy się przy kawie w biurze w budynku 16, zanim ktokolwiek inny się zjawił,

choć nasz zespół pracuje w budynku 43. Pewnego ranka, gdy rozmawialiśmy o SQL Server 2017, powiedział do mnie: „Czy wspominałem ci o naszych planach dla nowej wersji SQL Server, tej po SQL Server 2017?" Oczywiście chciałem się dowiedzieć więcej – „Jasne, słyszałem o tym, ale nie znam szczegółów". W efekcie zaprosił mnie na spotkanie następnego dnia, na którym miał przedstawić licznym członkom zespołu inżynierskiego plany tego projektu. Dopiero co spędziłem rok nad SQL Server 2016, teraz zostałem przydzielony do SQL Server 2017 i Linuksa, a tu Slava chce, abym zapoznawał się z wydaniem, które ma nastąpić *po* wydaniu, które jeszcze się nie ukazało? Ale oczywiście nie chciałem go zawieść, gdyż, no cóż, to jest Slava Oks. Może to brzmieć, jakby Slava był jakimś strasznym typem, ale w istocie jest jednym z najsympatyczniejszych ludzi, jakich kiedykolwiek poznałem w firmie Microsoft. Zatem podczas gdy zaczynałem pakować sobie do głowy najrozmaitsze szczegóły SQL Server 2017, jednocześnie rozpocząłem poznawanie tego, co miało stać się przyszłą wersją SQL Server, projektu o nazwie kodowej *Seattle*.

Projekt Seattle

W ciągu kilku godzin spotkania następnego dnia szybko zrozumiałem, że rozpoczynamy pracę nad najbardziej ambitnym usprawnieniem SQL Server, jakie widziałem do tej pory. Mówię to, mając świadomość, że w tym samym czasie wprowadzaliśmy na rynek SQL Server on Linux – coś, o czym nikt wcześniej nawet nie myślał, że jest możliwe.

Slava i jego zespół wybrał nazwę kodową „Seattle", gdyż nazwą kodową dla SQL Server 2017 było „Helsinki" i zespół szukał nowej „miejskiej" nazwy. O ironio, nikt w firmie Microsoft nie użył wcześniej nazwy Seattle, zatem szybko została przyjęta. Zapytałem Slava, kiedy zaczął planować Projekt Seattle. Byłem zdumiony słysząc, że sięgało to stycznia 2017. To, że tacy goście, jak Slava, Conor Cunningham i Travis Wright planowali Projekt Seattle, gdy pracowali nad budowaniem finalnych kawałków SQL Server 2017 i wersji dla Linuksa, pokazywało zarówno ich poświęcenie dla zespołu, jak i ich pragnienie, aby SQL Server nadal był czołową innowacją w branży bazodanowej.

Trudno było uwierzyć, że moglibyśmy tak szybko zaplanować coś większego po dostarczeniu tak wielu zadziwiających i innowacyjnych funkcji w SQL Server 2016 i SQL Server 2017.

W SQL Server 2016 wprowadziliśmy nowe możliwości diagnostyki wydajności poprzez Query Store. Dołączyliśmy nowe funkcje dla programistów, takie jak tabele czasowe i integracja z JSON. Poprawiliśmy naszą pozycję w dziedzinie zabezpieczeń za pomocą mechanizmu Always Encrypted, dynamicznego maskowania danych i zabezpieczeń poziomu wiersza. A ponadto wprowadziliśmy dwie innowacje wykraczające

poza „normalny" typ funkcjonalności relacyjnego systemu bazodanowego. Jedną była integracja z językiem R dla modeli uczenia maszynowego. Drugą integracja z systemami Hadoop poprzez funkcję nazwaną *Polybase* (która doprowadziła do czegoś jeszcze większego w wydaniu 2019, ale nie chcę wyprzedzać zdarzeń). Zbudowanie funkcji pozwalających na realizację nowych scenariuszy, takich jak uczenie maszynowe i Big Data, doprowadziło mnie (i innych) do wniosku, że SQL Server nie jest już jedynie silnikiem relacyjnej bazy danych, ale *platformą danych*.

Niemniej jednak, aby być nowoczesną i kompletną platformą danych, musieliśmy umożliwić stosowanie w systemach innych, niż tylko Windows Server. To doprowadziło nas do wydania SQL Server 2017, zawierającego wsparcie dla Linuksa i kontenerów Docker. Działanie na Linuksie oraz kontenery stanowiły wielką zmianę z punktu widzenia firmy Microsoft, ale SQL Server 2017 zawierał również inne możliwości, takie jak Adaptive Query Processing, automatyczne dostrajanie, grafowe baz danych, *bezklastrowe* grupy dostępności oraz integracja z językiem Python, uzupełniająca wsparcie dla języka R w Machine Learning Services.

Mając na uwadze te wszystkie innowacje, jak moglibyśmy w krótkim czasie zaplanować zbudowanie czegoś nowego, odmiennego i bardziej ekscytującego, niż wydania SQL Server 2016 i 2017? Zadałem sobie to pytanie, uważnie słuchając podczas pierwszego spotkania Projektu Seattle. Już w pierwszych minutach przedstawiono na nim koncepcję, która po bardziej publicznym ogłoszeniu została uznana za radykalną. Ta innowacja została rozpoczęta jako kamień węgielny projektu Seattle i otrzymała swą własną nazwę: *Aris*.

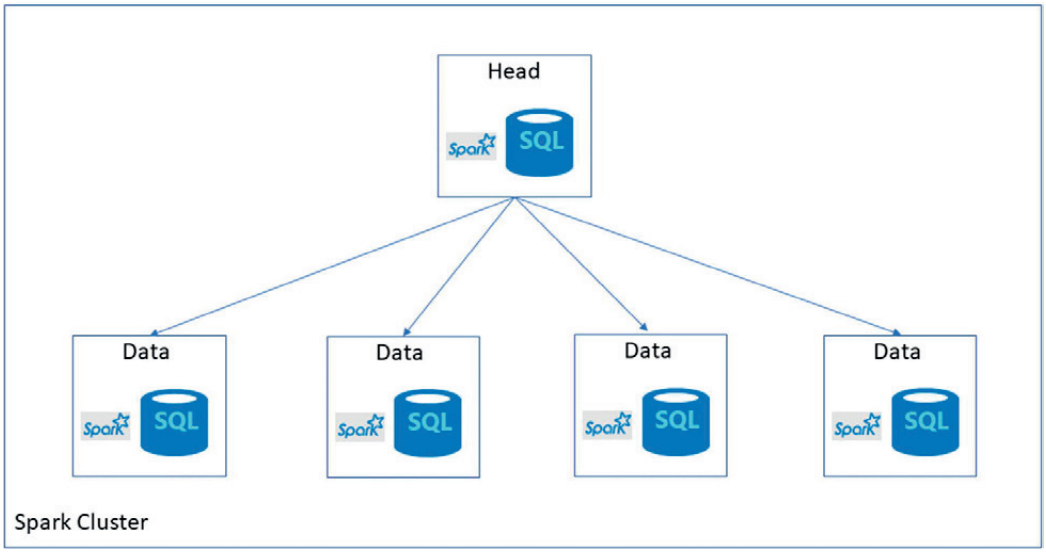
Projekt Aris

W styczniu 2017 roku Slava wraz z liderami zespołu inżynierskiego SQL Server otrzymali sugestię od Rohana Kumara, wiceprezesa do spraw Azure Data, aby przyjrzeć się temu, jak możnaby zintegrować SQL Server z *Big Data.* Termin ten, coraz częściej słyszany w branży i poza nią, jest luźnym określeniem systemów danych, które potrafią obsłużyć *wielkie* ilości danych, zazwyczaj poprzez rozproszoną, skalowalną platformę obliczeniową. Osobiście preferuję definicję Big Data zaproponowaną przez mojego kolegę Bucka Woody'ego: „Dowolne dane, których nie jesteśmy w stanie przetworzyć w czasie, w jakim byśmy chcieli, przy użyciu technologii, jaką mamy". Od wielu lat preferowanym wyborem systemu Big Data był Hadoop. Tak więc przez kilka miesięcy, wiosną i latem 2017 zespół przyglądał się pomysłom Travisa Wrighta, jak urzeczywistnić wizję integracji z Big Data. Latem 2017 zespół Azure Data miał w toku kilka projektów o takich nazwach kodowych, jak Polaris, Socrates czy Plato. Spytałem Slava, jak wybrali

nazwę *Aris*? Oto odpowiedź: Sokrates był nauczycielem Platona, zaś uczniem tego drugiego był Arystoteles. A ponieważ słowo *Aris* jest również częścią Polaris, nazwa zyskała akceptację wszystkich członków zespołu i kierownictwa.

Jako że integracja Big Data oznaczała konieczność zrobienia *czegoś* z Hadoop, Travis odbył szereg spotkań z zespołem, który wprowadził Polybase do SQL Server 2016 i do Azure Data Warehouse. Wizja rozszerzenia Polybase polegała na tym, aby pozwolić użytkownikom SQL Server na odpytywanie (i odbieranie) danych z systemu Hadoop wyłącznie przy użyciu języka T-SQL, tak dobrze znanego naszym istniejącym klientom. Co więcej, zamiast po prostu budowania prostego systemu ekstrakcji danych, Polybase mogłoby użyć siły przetwarzania rozproszonego, która jest już dostępna w Azure Data Warehouse i Analytics Platform System (wcześniej znanego pod nazwą Parallel Data Warehouse) w celu *zepchnięcia* (*pushdown*) obliczeń i partycjonowania przetwarzania zapytań, aby uzyskać skalowalną wydajność wobec wielkich zbiorów danych w docelowym systemie Hadoop. W rzeczywistości nigdy nie ujrzałem startu Polybase w SQL Server 2016 ani 2017, gdyż zintegrowanie systemów Big Data Hadoop z systemami relacyjnymi, takimi jak SQL Server, nie było łatwe. Polybase wymaga sporo pracy przy instalowaniu i konfiguracji, a modele zabezpieczeń w systemach Hadoop i SQL Server bardzo się różnią. Dodatkowo implementacja mechanizmu *pushdown* opierała się na koncepcji nazwanej MapReduce, wymagającej zainstalowania środowiska Java na tym samym komputerze, co SQL Server i usługi Polybase. Tym niemniej dostępna była architektura i koncepcje integracji SQL Server i systemów Big Data, pozwalające na zbudowanie czegoś większego (w tym rozszerzenie T-SQL o nazwie EXTERNAL TABLE). Gdybyśmy zdołali uprościć wdrożenie i konfigurowanie Polybase oraz dodać obsługę większej liczby źródeł danych, rozwiązanie to zapewne zostałoby lepiej przyjęte w branży. Co więcej, Travis bardzo szybko zrozumiał, że jeśli ktoś chce być poważnie traktowany w świecie Big Data, musi wziąć pod uwagę jeszcze inną technologię – *Spark*.

Uzbrojeni w tę wiedzę, Slava, Travis i kluczowa grupa członków zespołu budującego SQL Server on Linux mieli cel zbudowania prototypu integracji SQL Server z Big Data, uwzględniającej Spark. Zamknęli się na kilka dni w dużej sali konferencyjnej na czymś, co nazwano „Aris Hackathon". Do grupy tej należeli Slava Oks, Travis Wright, Scott Konersmann, Stuart Padley, Michael Nelson, Pranjal Gupta, Jarupat Jisarojito, Weiyun Huang, George Reynya, David Kryze, Umachandar Jayachandran (UC) oraz Sahaj Saini. Gdy zakończyli pracę, mieli *działający klaster*, łączący istniejącą funkcjonalność Polybase ze Spark. Rysunek 1-1 prezentuje szkicowy diagram tego klastra.



Rysunek 1-1 *Pierwszy klaster Aris*

W tym prototypie zbudowali klaster Hadoop zawierający komponenty dla Apache Spark i HDFS, ale również powiązane z SQL Server Polybase. Wykorzystali Spark do skierowania strumienia danych do węzłów *Data* i użyli Polybase do złączenia danych przechowywanych w węźle *Head* (w SQL Server) z danymi wciągniętych za pomocą Spark do HDFS. Ideą tego prototypu było wykazanie, że możliwe jest zintegrowanie ze sobą komponentów Spark, Hadoop i SQL Server.

Mniej więcej w tym samym czasie Travis rozmawiał z inżynierami, którzy dołączyli do zespołu z firmy Metanautix przejętej niedawno przez Microsoft. Jako część tego przejęcia nasz zespół zdobył technologię łączenia się przez ODBC z szeregiem źródeł danych, w tym ORACLE, SQL Server, Teradata i MongoDB. Zespół uważał, że jeśli zdołamy zintegrować tę technologię z projektem Aris, będziemy mieli całkiem wciągającą historię *wirtualizacji danych*. SQL Server mógłby być teraz hubem dla uzyskiwania dostępu do danych na różnych platformach i systemach, bez konieczności przenoszenia tych danych do SQL Server za pomocą takich technik, jak ekstrakcja, transformacja i ładowanie (ETL).

Zanim jednak mogliśmy dostarczyć oprogramowanie, którego nasi kliencie mogliby użyć i wypróbować, musieliśmy wybrać platformę, na której miałyby działać wszystkie te komponenty. Potrzebowaliśmy platformy, która pozwoliłaby na łatwe wdrażanie całego oprogramowania, w tym Polybase, Hadoop i Spark, zapewniała możliwości zarządzania i zabezpieczenia oraz elastyczne skalowanie i wysoką dostępność. Logicznym

wyborem wydawały się kontenery, biorąc pod uwagę łatwość ich instalowania, a wraz z wydaniem SQL Server 2017 zapewniliśmy wsparcie dla SQL Server w kontenerach. Kolejną naturalną decyzją był wybór Kubernetes jako platformy do zbudowania klastra, w którym działały te kontenery. Kubernetes właśnie szybko nabierały tempa jako platforma dla przetwarzania rozproszonego i skalowania wydajności. Z naszych doświadczeń wiedzieliśmy już, że dla Kubernetes i systemów Hadoop preferowanym systemem operacyjnym jest Linux, a ponieważ SQL Server już był obsługiwany na Linuksie, pasował dobrze do tego wyboru.

W ten sposób wraz z końcem roku 2017 nasz zespół wyruszył w podróż, której celem było zbudowanie klastra *Aris*, realizującego nie tylko wizję wirtualizacji danych, ale także integracji z technologiami Big Data, takimi jak Spark i HDFS. Od samego początku zespół zdecydował, że wszystko to musi zostać umieszczone „w jednym pudełku". Innymi słowy, jeśli ktoś kupi SQL Server, będzie mógł zainstalować wszystkie te komponenty w ramach swojej licencji (bez konieczności wiedzy, co jest nowym składnikiem, po prostu wszystko ma być dołączone do SQL Server). Finalny produkt, który mamy dziś w rękach, to SQL Server 2019, zaś to, co nazywamy *SQL Server Big Data Clusters* (*klastrami Big Data*), zawiera znacznie więcej możliwości, niż wczesne prototypy Aris, ale wizja i koncepcje są te same: dostarczenie łatwej do wdrożenia platformy wirtualizacji danych z wbudowanymi skalowalną wydajnością, zabezpieczeniami i możliwościami zarządzania.

Seattle staje się SQL Server 2019

Choć koncepcja Aris i klastrów Big Data była czymś wielkim, innowacyjnym i, powiedzmy to szczerze, dość onieśmielającym, każde główne wydanie SQL Server zawiera usprawnienia w wielu obszarach platformy. Obejmują one wydajność, zabezpieczenia i dostępność, czyli te trzy, które Conor Cunningham często określa mianem „chleba powszedniego SQL Server". Nasz zespół dodatkowo przygotował SQL Server dla Linuksa w wydaniu SQL Server 2017. Choć ten produkt był zadziwiający, pozostało nadal kilka funkcjonalności, które były dołączane do SQL Server dla Windows, a więc musiały zostać również dołączone do wersji linuksowej. Wiedzieliśmy też, że kontenery są czymś wielkim, w tym sensie, że stanowiły przyszłościowy kierunek wdrażania i uruchamiania aplikacji, w tym SQL Server. Było zatem sporo pracy do zrobienia, o której wiedzieliśmy, w tym zbadanie nowych scenariuszy z użyciem klastrów Kubernetes (nie tylko dla Big Data Clusters).

Tworzenie tak wielkiego produktu, jak SQL Server, wymaga udziału wielu zespołów deweloperskich. Nasz zespół Enterprise (inaczej *Tiger Team*) miał całą stertę nowych funkcjonalności, które chcieliśmy dodać w nowym wydaniu, o prawdziwej wartości dla klientów. Nasi koledzy, którzy budowali nowe funkcje wydajności, dostępności i zabezpieczeń dla Azure SQL Database, również chcieli ujrzeć efekty swojej pracy w Projekcie Seattle, gdyż silniki realizujące usługę Azure i SQL Server są tym samym. W 2017 roku miałem świadomość, że rozpoczynamy drogę w stronę historycznego wydania.

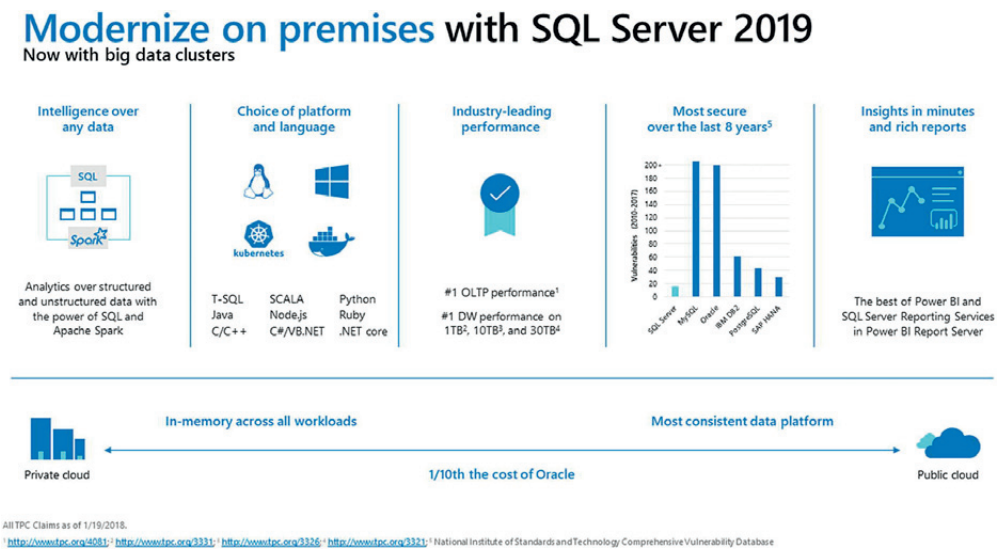
Zanim rok 2017 się skończył, byliśmy już przygotowani do pracy nad kolejnym wydaniem SQL Server, czyli SQL Server 2018. Wydawało się to sensowne, przynajmniej dla mnie. Wypuściliśmy dwie duże wersje SQL Server w dwóch poprzednich latach, SQL Server 2016 oraz SQL Server 2017, zatem czemu nie SQL Server 2018?

Conor Cunningham, nasz architekt produktu i wydania, powiedział mi, że przy naszych umiejętnościach pracy *agile* moglibyśmy wydawać kolejną wersję SQL Server co miesiąc, gdybyśmy tylko zechcieli. I możemy to robić z utrzymaniem wysokiej jakości. Naturalnie nie będziemy tego robić, gdyż chcemy wypuszczać kolejne wydania SQL Server, które będą zarówno wysokiej jakości, jak i będą wartościowe dla naszych klientów. W pierwszych miesiącach roku 2018 musieliśmy zdecydować, czy chcemy wydać nową wersję w tym roku. Po przyjrzeniu się tym możliwościom, które byliśmy w stanie umieścić w tym wydaniu, w tym Big Data Clusters, wiosną 2018 zdecydowaliśmy, że możemy opublikować wersję wstępną SQL Server vNext w kalendarzowym roku 2018. (Gdy nie znamy nazwy oficjalnej kolejnego wydania, nazywamy ją „vNext", pomimo że dysponowaliśmy nazwą projektu, taką jak Seattle). Jak Czytelnik zapewne zauważył, często staramy się przedstawiać nowe główne wydania na wielkich wydarzeniach. Po zajrzeniu do kalendarza stwierdziliśmy, że jednym z największych wydarzeń klienckich (dla oprogramowania Microsoft) będzie Microsoft Ignite (obecnie odbywające się w Orlando, z około 30 tysiącami uczestników). Tak więc latem 2018 kierownictwo zdecydowało się zaprezentować wstępną wersję SQL Server vNext na konferencji Microsoft Ignite i nazwać ją SQL Server 2019, rozumiejąc przez to, że wydanie to uzyska status GA (co oznacza *General Availability* – ogólna dostępność) kiedyś w roku 2019.

Miało to sens dla wszystkich członków zespołu. Zapewniało to dłuższy pas lądowania klastrów Big Data, a także więcej czasu na zrealizowanie zmian w „jądrze" SQL Server, wszystkich opartych na informacjach zwrotnych od klientów. Moje zadanie? Wykorzystać wykonaną przeze mnie pracę do propagowania i prezentowania SQL Server 2016 i 2017 oraz pokazać naszym klientom, całej branży i społeczności, że w wydaniu SQL Server 2019 naprawdę zbudowaliśmy *nowoczesną platformę danych*.

Modernizowanie bazy danych przy użyciu SQL Server 2019

Rysunek 1-2 jest moim głównym „lokującym" diagramem dla wystąpień o SQL Server 2019. Zbudowany przez moją koleżankę z działu marketingu firmy Microsoft, Debbi Lyons (ktoś mógł zauważyć, że niekiedy pojawiam się razem z Debbi podczas wykładów na temat SQL Server), przedstawia pełny obraz nowoczesnej platformy danych SQL Server 2019.



Rysunek 1-2 *Modernizowanie przy użyciu SQL Server 2019*

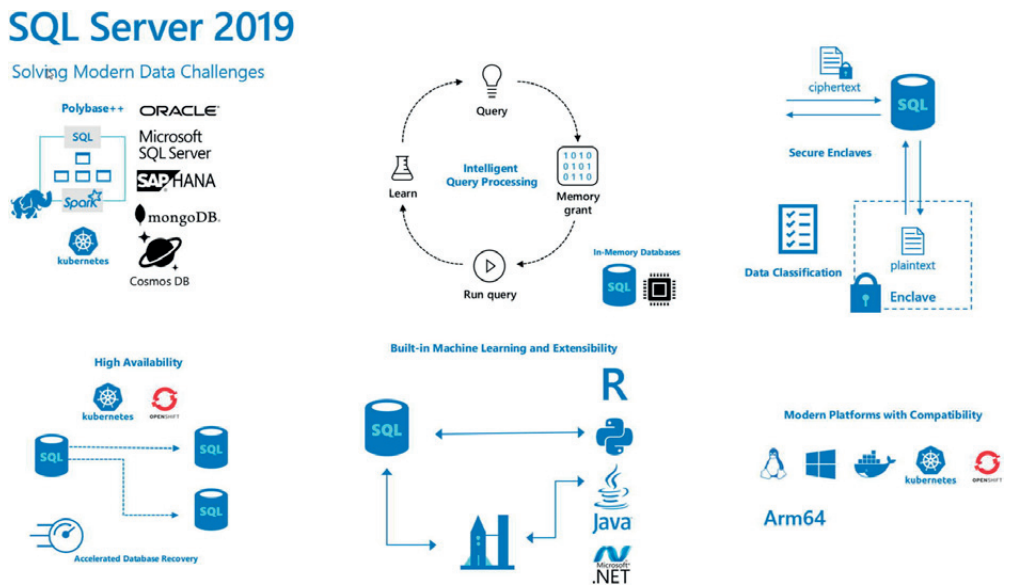
Jeśli ktoś uczestniczył w moich wystąpieniach na temat SQL Server 2016 lub 2017, mógł zauważyć, że slajd wyglądał podobnie, ale z kluczowymi różnicami:

- Zintegrowane rozwiązanie wirtualizacji danych (Data Virtualization), łączące Spark, HDFS i SQL Server w innowacyjny sposób (zasadniczo „SQL Server spotyka Big Data")
- Nowe możliwości wyboru platformy, obejmujące Windows, Linux, kontenery oraz Kubernetes

SQL Server od dekady zajmuje czołowe miejsce w branży bazodanowej pod względem wydajności i jako najmniej podatna platforma danych. Dysponując licencją na SQL

Server, klienci mają dostęp do usług Business Intelligence, takich jak Power BI Report Server. Dodatkowo dzięki nowej usłudze Azure SQL Database Managed Instance dostępna jest taka sama funkcjonalność w przypadku instalacji SQL Server w chmurze prywatnej i Azure w chmurze publicznej. Ta spójność nie ogranicza się do zgodności danych. Nasze umiejętności w zakresie T-SQL mają zastosowanie zarówno do SQL Server, jak i Azure, zaś narzędzia działają jednakowo w lokalnej instalacji i w usługach Azure Data.

Inny zestaw możliwości, który często ginie w tłumie innych możliwości, to fakt, że SQL Server (i Azure) udostępniają funkcjonalności pamięciowe, które pozwalają zmaksymalizować wykorzystanie zasobów obliczeniowych, w tym In-Memory OLTP oraz Columnstore Indexes. Wszystkie te elementy należą do SQL Server 2019. Rysunek 1-3 przedstawia bardziej szczegółowy obraz kluczowych nowych funkcjonalności, unikatowych dla wydania SQL Server 2019.



Rysunek 1-3 *Kluczowe nowe funkcjonalnosci SQL Server 2019*

Zamierzam wykorzystać ten diagram (podążając od lewej do prawej, od lewego górnego rogu), aby naszkicować główne nowe funkcje SQL Server 2019, co można będzie wykorzystać jako plan dla lektury reszty tej książki. Czytając o tych nowych możliwościach trzeba pamiętać, że to SQL Server zasila Azure SQL Database, co oznacza, że wiele możliwości przedstawionych w tej książki działa identycznie w Azure SQL Database. Co więcej, wszystko, co pokażę w książce, można zrealizować w Azure, bez względu

na to, czy będzie to SQL Server w maszynie wirtualnej Azure, czy kontenery i Kubernetes w chmurze.

Wirtualizacja danych

Wcześniej w tym rozdziale wspomniałem o tym, że źródłem koncepcji wirtualizacji danych był projekt Aris. SQL Server 2019 stanowi realizację tej wizji poprzez dwie konkretne funkcjonalności:

- **Polybase w SQL Server 2019** Nazywam to Polybase++, gdyż rozszerzyliśmy funkcjonalność Polybase dostarczanej z SQL Server 2016 (więcej informacji na temat Polybase zawiera strona `https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-guide?view=sql-server-2017`), aby dostarczyć łączniki do różnych źródeł danych, w tym Oracle, SQL Server, MongoDB (CosmosDB) oraz Teradata. Przy czym można łączyć się z tymi źródłami danych bez instalowania żadnego oprogramowania klienckiego; SQL Server zawiera wbudowane wszystko, czego potrzebujemy. Dodatkowo można łączyć się z jeszcze innymi źródłami, takimi jak SAP HANA, instalując swój własny sterownik ODBC. Nową wersję Polybase w SQL Server 2019 omawiam w rozdziale 9.
- **Big Data Clusters** Jak wspomniałem w przedstawieniu naszej wizji projektu Aris wcześniej w tym rozdziale, zdecydowaliśmy się na zbudowanie kompletnego rozwiązania obejmującego SQL Server z nową funkcjonalnością Polybase, HDFS, Spark oraz inne komponenty na użytek zarządzania, zabezpieczeń i dostępności. Tematyka ta obejmuje znacznie więcej treści, niż mógłbym zamieścić w tym miejscu – więcej informacji na ten temat zawiera rozdział 10.

Uwaga Oryginalnie chciałem zamieścić te tematy w drugim i trzecim rozdziale książki. Jednak później zdecydowałem, że pomocne będzie przedstawienie więcej informacji na temat kontenerów i Kubernetes, zatem konieczne jest wcześniejsze zaprezentowanie tych treści. Stąd „krzykliwe rozpoczęcie" od tych innowacji. Jeśli Czytelnik nie potrzebuje takiej pomocy, może od razu zagłębić się w treść rozdziału 9.

Wydajność

Przy każdym wydaniu SQL Server zajmujemy się wydajnością. Zawsze. Jednak samo zapewnienie, że zapytania użytkowników będą wykonywane szybko, nie wystarczy. Musimy pracować nad tym, aby silnik SQL Server stał się bardziej inteligentny i dostosowywał się do obciążenia, inwestycji w sprzęt czy wzorców złożonych zapytań. Rozdział 2 zawiera wyczerpujący przegląd możliwości wydajnościowych SQL Server 2019, obejmując między innymi (ale nie tylko):

- Mechanizm *Intelligent Query Processing* (inteligentnego przetwarzania zapytań), stanowiący rozszerzenie mechanizmu Adaptive Query Processing wprowadzonego w SQL Server 2017.
- Dostęp do planów zapytań *gdziekolwiek i kiedykolwiek* to potrzebne dzięki ulepszeniom Lightweight Query Profiling (lekkie profilowanie zapytań), Last Execution Plan (ostatni plan wykonania) oraz Query Store (magazyn zapytań).
- Rodzina możliwości zapewniających prawdziwą *pamięciową bazę danych* (*in-memory database*), w tym odciążone mechanizmy I/O (wejścia/wyjścia) oraz Hybrid Buffer Pool dla trwałego schematu optymalizowanej pamięciowo bazy `tempdb`. Połączenie tych technologii z wbudowanymi mechanizmami Columnstore Indexes oraz In-Memory OLTP zapewnia bardzo skuteczne rozwiązanie pamięciowej bazy danych.

Zabezpieczenia

SQL Server jest nie tylko najmniej podatnym produktem bazodanowym w branży, jeśli spojrzymy na wyniki z ostatniej dekady, ale zawiera też szeroki zakres funkcji i narzędzi mających na celu zaspokojenie współczesnych wymagań zabezpieczeń. Obejmują one następujące usprawnienia wprowadzone w SQL Server 2019:

- **Always Encrypted with Secure Enclaves** SQL Server 2016 wprowadził nowy całościowy system zabezpieczeń dla aplikacji danych o nazwie *Always Encrypted* (zawsze szyfrowane). Choć system ten zapewnia szyfrowanie danych podczas przechowywania, w pamięci i przy przesyłaniu przez sieć, istniało kilka ograniczeń, z których najważniejszym było tzw. *bogate przetwarzanie* (*rich computing*). W rozdziale 3 pokażę, jak funkcja Always Encrypted przy użyciu koncepcji nazywanej *Secure Enclaves* (bezpieczne enklawy) umożliwia bogate przetwarzanie i inne interesujące scenariusze zabezpieczeń.

- **Data Classification and Auditing** Ogólne rozporządzenie o ochronie danych (RODO, ang. *General Data Protection Regulation* – GDPR) obowiązuje w krajach Unii Europejskiej od maja 2018. Od tego czasu rozmawiałem z wieloma klientami zlokalizowanymi w UE oraz z firmami, które prowadzą interesy z klientami w UE. Nowa wbudowana funkcja Data Classification and Auditing (klasyfikacja danych i inspekcja), w połączeniu z innymi narzędziami, mogą być bardzo pomocne w zapewnieniu zgodności z RODO i innymi regulacjami, którym muszą sprostać przedsiębiorstwa.

Te nowe funkcje i inne zagadnienia związane z zabezpieczeniami omawiam w rozdziale 3.

Dostępność krytyczna

Szybkość i bezpieczeństwo są ważne, ale firmy, których działanie opiera się na SQL Server, potrzebują, aby ich platforma danych była dostępna bez przerwy. SQL Server 2019 zawiera nowe możliwości zapewniania wysokiej dostępności danych, w tym:

- Funkcje Resumable Online Create Index i Clustered Columnstore Online Create Index, uzupełniające wymogi dostępności indeksów bez przerywania pracy.
- Ulepszenia flagowej funkcjonalności HADR, czyli Always On Availability Groups, w tym zwiększenie liczby replik oraz przekierowywanie głównego połączenia.
- Wyobraźmy sobie świat, w którym wycofywanie transakcji następuje natychmiast, zaś przywracanie i przycinanie dzienników nie zależy od wielkich lub długotrwałych transakcji. Witamy w nowym świecie Accelerated Database Recovery!

Te i inne krytyczne rozwiązania dostępności omówię w rozdziale 4.

Nowoczesna platforma programistyczna

Jak dotąd, wszystkie nowe rzeczy, o których wspominałem jako elementach SQL Server 2019, dotyczyły tylko administratorów baz danych lub inżynierów IT. Jesteśmy jednak przekonani, że programiści są ważnym czynnikiem sukcesu SQL Server, zatem zainwestowaliśmy również w następujące nowe funkcje:

- W SQL Server 2016 wprowadziliśmy nową platformę Machine Learning (uczenia maszynowego) przy użyciu języka R. W SQL Server 2017 rozszerzyliśmy ten model, dodając wsparcie dla języka Python. Korzystając z tej samej infrastruktury, pozwalamy teraz programistom rozszerzać język T-SQL przy użyciu klas Java.

W istocie zbudowaliśmy rozszerzalne SDK, pozwalając na dołączanie jeszcze innych języków do świata SQL Server.

- Rozszerzyliśmy możliwości grafowej bazy danych, wprowadzonej po raz pierwszy w SQL Server 2017, o nowe funkcje, takie jak ograniczenia krawędzi i wsparcie dla klauzuli MERGE.
- Chcemy zachęcać programistów do korzystania z Unicode, zatem dodaliśmy nowe sortowania UTF-8, pomagające programistom obsługiwać dane UTF-8 bez narzutu związanego z typami danych Unicode.

Zorientowane na programistów funkcje zawarte w SQL Server 2019 omawiam w rozdziale 5.

Inwestowanie w platformę wyboru

Udało się nam przygotować SQL Server dla Linuksa w wydaniu SQL Server 2017, ale mieliśmy kilka funkcji silnika, których nie udało się nam zrealizować w tamtym wydaniu. Chcemy jednak, aby nasi użytkownicy mieli swobodę wyboru systemu operacyjnego dla SQL Server, bez martwienia się o dostępność funkcji lub kompatybilność. Poprawiliśmy to obecnie w wydaniu SQL Server 2019, dodając funkcjonalności Replication, Change Data Capture (CDC), Distributed Transactions (DTC), Machine Learning oraz Polybase do SQL Server dla Linuksa.

Zainwestowaliśmy również w technologię kontenerów, dołączając nowy rejestr kontenerowy, wsparcie dla Red Hat Enterprise Linux (RHEL) i rozszerzając obsługę Kubernetes, w tym OpenShift. A choć nie omawiam tego w tej książce, rozszerzyliśmy listę platform dla SQL Server o procesory Arm przy wykorzystaniu Azure SQL Database Edge. Więcej informacji na temat rozwiązań Azure SQL Database Edge zawiera strona `https://azure.microsoft.com/en-us/services/sql-database-edge/`.

Warto tu zatrzymać się na chwilę i przyjrzeć wszystkim ikonom platform, gdyż SQL Server nie jest już po prostu platformą wyboru. To *platforma wyboru z zapewnieniem zgodności*. Możemy wykonać kopię bazy danych na dowolnej z tych platform i odtworzyć ją na innej, bez konieczności dokonywania jakichkolwiek zmian.

Zagadnieniom dotyczącym ulepszeń SQL Server dla Linuksa, kontenerów SQL Server oraz Kubernetes poświęciłem rozdziały 6, 7 i 8 tej książki.

Oprócz tych głównych obszarów istnieją jeszcze inne innowacje, o których warto wspomnieć. Znajdą się one w różnych rozdziałach książki.

Azure Data Studio

SQL Server Management Studio (SSMS) przez wiele lat stanowiło podstawowy interfejs użytkownika SQL Server. W zeszłym roku zdecydowaliśmy się na zbudowanie nowego narzędzia eksploracji danych, zapewniającego rozszerzalność i nowe możliwości, pod roboczą nazwą SQL Operations Studio. Narzędzie to zostało oficjalnie udostępnione we wrześniu 2018 pod nazwą Azure Data Studio (ADS).

Azure Data Studio dysponuje kilkoma innowacyjnymi (lub nowymi dla świata SQL Server) technologiami, takimi jak Notebooks, obsługa Big Data Clusters, External Data Wizards oraz eksploracją usług SQL Server, HDFS i innych usług danych Azure.

Nie utworzyłem odrębnego rozdziału poświęconego Azure Data Studio. Zamiast tego prezentuję wykorzystanie tego narzędzia (obok SSMS i innych) w różnych miejscach tej książki.

Głosy klientów

Jako że moja kariera zawodowa rozpoczęła się od pracy w pomocy technicznej dla klientów, zawsze cieszy mnie, gdy nasz zespół inżynierski dodaje do nowych wydań takie funkcjonalności, które można bezpośrednio powiązać z informacjami od klientów lub trendami w problemach zgłaszanych do naszego zespołu CSS.

To wydanie nie jest wyjątkiem i zawiera szereg usprawnień silnika bazodanowego, w tym (ale nie tylko):

- Lepszy sposób przycinania komunikatu błędu z kontekstowym działaniem. Jest to funkcjonalność, która pojawiała się najczęściej w zgłoszeniach klienckich (obejmujących tysiące głosów).
- Nowe dynamiczne obiekty zarządzania, pozwalające uzyskiwać wgląd w wewnętrzne elementy nagłówków stron baz danych (tak, teraz każdy może być Paulem Randalem!). Te instrukcje mogą znacząco pomóc w rozwiązywaniu problemów dotyczących rywalizacji o uchwyt strony.
- Usprawnienia skalowalności silnika, obejmujące współbieżne aktualizacje PFS, równoległe wstawianie wsadowe i pośrednie punkty kontrolne.

Więcej szczegółów tej kolekcji usprawnień zamieściłem w rozdziale 11.

Z punktu widzenia struktury, rozdziały książki są w znacznym stopniu niezależne od siebie. Polecam jednak przeczytanie rozdziałów 7 i 8 jako informacji o podstawach, zanim zdecydujemy się zagłębić w rozdziały 9 i 10, dotyczące wirtualizacji danych i Big Data Clusters.

Zaczynamy pracę z SQL Server 2019

W tym miejscu przedstawię kilka źródeł pomagających w instalacji i konfigurowaniu SQL Server 2019, aby móc praktycznie poznawać nowe funkcje i wypróbować przykłady zawarte w pozostałych rozdziałach.

Pobieranie SQL Server 2019

Aby pobrać i wypróbować SQL Server 2019, należy przejść do strony `www.microsoft.com/en-us/sql-server/sql-server-2019#Install`.

Instalowanie SQL Server 2019

Instrukcje dotyczące wdrożenia SQL Server 2019 w systemie Windows zawiera strona `https://docs.microsoft.com/en-us/sql/database-engine/install-windows/installation-for-sql-server?view=sql-server-ver15`.

W przypadku SQL Server 2019 dla Linuksa należy przejść do strony `https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-overview?view=sql-server-ver15`.

Aby nauczyć się, jak wdrożyć SQL Server w kontenerze, należy przejść do strony `https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-linux-ver15&pivots=cs1-bash`.

Migracja do SQL Server 2019

Rozdział 11 zawiera omówienie migracji i narzędzi wspomagających przejście do Server 2019 z wcześniejszych wydań SQL Server lub produktów bazodanowych innych dostawców.

Co nowego w SQL Server 2019

Całość nowych funkcjonalności zawartych w SQL Server 2019 przedstawia strona `https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-ver15?view=sqlallproducts-allversions`.

Pobieranie kodu książki i przykładowych baz danych

Jeśli Czytelnik chce wypróbować wszystkie przykłady zawarte w tej książce, zalecam sklonowanie repozytorium GitHub dla tej książki, zgodnie ze wskazówkami zawartymi we wstępie.

Wskazówka Użytkownicy Windows powinni posłużyć się poniższą składnią polecenia **git** przy klonowaniu repozytorium, aby uniknąć problemów związanych z zakończeniami wierszy (CRLF kontra LF) w linuksowych skryptach:

```
git clone --config core.autocrlf=false https://github.com/microsoft/
sqlworkshops.git
```

Dodatkowo trzeba pobrać przykładowe bazy danych: WideWorldImporters ze strony `https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0` oraz WideWorldImportersDW ze strony `https://github.com/Microsoft/sql-server-samples/releases/download/wide-world-importers-v1.0/WideWorldImportersDW-Full.bak`. Kod dla książki zawiera przykłady odtwarzania tych kopii zapasowych w Windows, Linuksie, kontenerach i Kubernetes.

SQL Server Workshops

Choć dołączyłem do książki wiele praktycznych ćwiczeń, warto zajrzeć na stronę `http://aka.ms/sqlworkshops`, aby znaleźć więcej bezpłatnych szkoleń na temat SQL Server (głównym animatorem tej strony jest mój przyjaciel i współpracownik Buck Woody, jeden z najlepszych nauczycieli, jakich znam).

Czy to jest SQL Server naszych dziadków?

Pisanie tej książki podobało mi się nie tylko dlatego, że lubię technologię (OK, mam skrzywienie w stronę SQL Server), ale także dlatego, że nasz zespół inżynierski wprowadza innowacje w tempie niespotykanym w żadnym konkurencyjnym produkcie bazodanowym w branży. I przyznajmy to, uczenie się nowych rzeczy *jest* fajne.

Być może najlepiej przedstawi to ten cytat z *ITProToday*: „Nigdy nie spodziewałem się, że pewnego dnia będę dyskutować o funkcjach nowego wydania Microsoft SQL Server, w tym samym zdaniu używając słów Linux, Oracle czy Apache Spark, ale taki jest właśnie ten nowy, wspaniały świat. Microsoft rozwija SQL Server w tempie, z którym

nie może się mierzyć żaden z jego konkurentów" (www.itprotoday.com/sql-server/polybase-expansion-big-clusters-are-key-features-new-sql-server-2019).

Pamiętam, co mój kolega Travis Wright powiedział o SQL Server 2019: „To nie jest SQL Server twojego dziadka". Rozumiał przez to, że produkt wyewoluował już z bycia silnikiem relacyjnej bazy danych i stał się prawdziwą platformą danych, zawierającą takie przełomowe technologie, jak Spark, HDFS, Notebooks, Polybase, R, Python, Java, Linux, kontenery i Kubernetes.

Pamiętam, że zamieściłem ten cytat na Twitterze. Mój kolega Pedro Lopes zauważył to i skomentował, że jednak SQL Server 2019 naprawdę *jest* SQL Serverem dziadka. Zatem kto ma rację? Obydwaj. SQL Server 2019 nadal jest silnikiem relacyjnej bazy danych, którą znamy i lubimy od lat, o skalowalnej wydajności, zabezpieczeniach i wysokiej dostępności. I jak zobaczymy w tej książce, zawiera usprawnienia w tych wszystkich kluczowych obszarach. Ale SQL Server 2019 to znacznie więcej: to jedna z najpopularniejszych platform bazodanowych na planecie i najnowszy gracz na rynku. Jest jednym i drugim.

Rozdział 2

Inteligentna wydajność

Wydajność jest krytyczna dla działań dowolnej platformy danych. Rozdział ten zawiera informacje o tym, jak SQL Server 2019 pomaga zwiększyć wydajność zapytań bez zmian w aplikacji. To jeden z najdłuższych rozdziałów w książce, zawierający mnóstwo przykładów, zatem zapinamy pasy i do przodu.

Dlaczego inteligentna wydajność?

Moim zdaniem, najważniejszą rzeczą, jaką Czytelnik może (czy też powinien) wynieść z tej książki, jest zrozumienie, **dlaczego** nowe możliwości SQL Server 2019 mogą być korzystne lub pomocne w rozwiązywaniu określonego problemu lub wyzwania. W przypadku wydajności motywem przewodnim jest umożliwienia podniesienia wydajności dla rozmaitych obciążeń, często **bez** dokonywania jakichkolwiek zmian w aplikacjach ani zapytaniach.

We wrześniu 2018 przygotowywałem prezentację na konferencję Microsoft Ignite w Orlando na Florydzie. Do tego czasu powszechnie znane były tylko bardzo ogólne plany dotyczące SQL vNext. Wraz z kolegą Amitem Banerjee mieliśmy zaprezentować wstępną wersję SQL Server 2019 na tej konferencji. W trakcie budowania wystąpienia zdawaliśmy sobie sprawę, że musimy podkreślić nowe usprawnienia wydajnościowe. Amit był twórcą nowego terminu *Intelligent Database* (inteligentna baza danych). Koncepcja polegała na tym, że SQL Server zawiera możliwości włączające inteligencję do silnika, pozwalającą mu na wykrywanie, dostosowanie i udostępnianie wglądu w dane, jak nigdy wcześniej.

Wykorzystałem ten termin, aby skupić się na wydajności, używając zwrotu *Intelligent Performance* (inteligentna wydajność). Obejmuje ona następujące ulepszenia obecne w SQL Server 2019:

- Intelligent Query Processing (inteligentne przetwarzanie zapytań)
- Lightweight Query Profiling (lekkie profilowanie zapytań)
- In-Memory Database (baza danych w pamięci)
- Last-Page Insert Contention (rywalizacja o wstawienia na ostatniej stronie)

Każdy z tych obszarów obejmuje inteligencję wbudowaną w silnik SQL Server, pomagającą uzyskanie lepszej wydajności, w wielu przypadkach bez jakichkolwiek zmian. W części sytuacji SQL Server zapewnia wgląd w wydajność zapytań na nigdy dotąd nie widzianym poziomie. W innych wbudowane możliwości pozwalają automatycznie wykorzystać innowacyjne funkcje sprzętu.

Podczas pisania książki trzeba podejmować najrozmaitsze decyzje. Jedną z nich jest uporządkowanie treści w rozdziałach. Ten rozdział jest bardzo długi, głównie z powodu przykładów zawierających wiele wizualnych przedstawień. Jestem wzrokowcem, zatem wydawało mi się, że będzie to dobry sposób zaprezentowania tych nowych funkcji. Każdy podrozdział jest w istocie oddzielnym rozdziałem i można go tak traktować. Zdecydowałem się zamieścić je razem, gdyż chciałem wspólnie przedstawić wszystkie szczegóły i rozległość *Intelligent Performance* w SQL Server 2019.

Każdy podrozdział tego rozdziału zaczyna się od wymagań wstępnych dla zamieszczonych przykładów. Jako wspólne wymagania wysokiego poziomu, musimy mieć:

- działającą instalację SQL Server 2019 w systemie Windows lub Linux
- SQL Server Management Studio (SSMS) 18.0 lub późniejsze
- Azure Data Studio (dowolny system operacyjny, ale wymagana minimalna wersja do 1.7.0)

Wiele przykładów wykorzystuje SSMS do wyświetlania planów zapytań, ale w trakcie przechodzenia od przypadku do przypadku można też wykorzystać Azure Data Studio (jedynie trzeba będzie oglądać plany w formie XML), używając nowego rozszerzenia SentryOne Plan Explorer. Więcej informacji na temat tego rozszerzenia dostępnych jest pod adresem `https://cloudblogs.microsoft.com/sqlserver/2019/07/11/the-july-release-of-azure-data-studio-is-now-available`.

Intelligent Query Processing

W wydaniu SQL Server 2014 nasz zespół inżynierski podjął ważką decyzję o wprowadzeniu nowego zestawu kodu dla procesora zapytań podejmującego decyzje związane z szacowaniem kardynalności (*cardinality estimation* – CE) zapytania. Nowy „model CE" jest stosowany, gdy baza danych działa na poziomie zgodności 120 lub późniejszym (120 jest ustawieniem domyślnym dla SQL Server 2014). Wszelkie szczegóły jego działania i powody wprowadzenia tej zmiany można znaleźć w dokumentacji dostępnej pod adresem `https://docs.microsoft.com/en-us/sql/relational-databases/performance/cardinality-estimation-sql-server`.

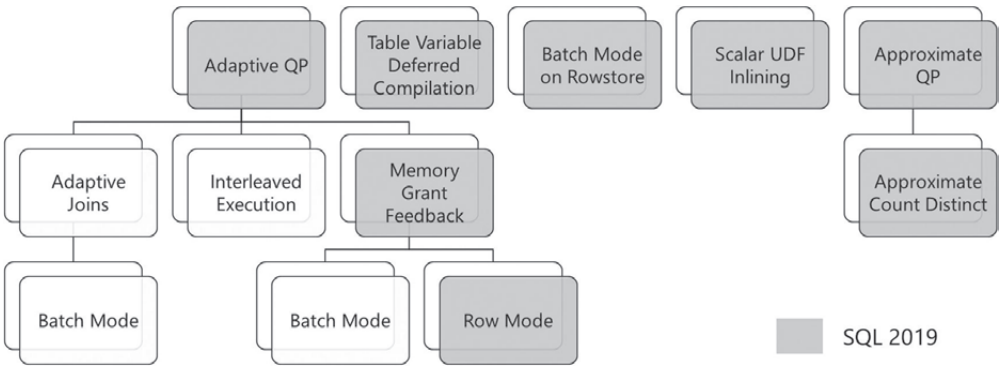
Wielu użytkowników kwestionowało słuszność tej decyzji. Jednym z głównych zarzutów stawianych temu podejściu było to, że była to obszerna, nieelastyczna zmiana. Gdy zespół kończył budowanie SQL Server 2016 i zaczynał planowanie wydania SQL Server 2017, wszyscy byli zgodni, że potrzebujemy nowego sposobu tworzenia funkcjonalności przetwarzania zapytań. Jak powiedział Joe Sack, jeden z czołowych menedżerów programu dla komponentu Query Processor (QP), „Zespół zrozumiał, że rozwiązania typu 'pasuje do wszystkiego' nie było tym, co powinniśmy robić. Zamiast tego powinniśmy zainwestować w funkcjonalności, które będą mogły dostosować się do szerokiego zakresu obciążeń, które występują w ekosystemie SQL Server (wielkie, małe, OLTP, hybrydowe, hurtownie danych)".

W ten sposób narodziła się nowa *rodzina funkcjonalności* w SQL Server 2017, o nazwie **Adaptive Query Processing (AQP)**. Pomysł polegał na wbudowaniu w procesor zapytań zdolności do *adaptacji* w miarę wykonywania zapytania (lub przed jego *ponownym* wykonaniem), aby zapewnić szybsze wykonywanie bez jakiejkolwiek interwencji użytkownika ani zmian w aplikacji.

Uwaga Przykłady dotyczące SQL Server 2017 i AQP dostępne są pod adresem `https://github.com/Microsoft/bobsql/tree/master/demos/sqlserver/aqp`.

Gdy zespół wypuścił SQL Server 2017 i AQP, istniał już obszerny rejestr nowych funkcjonalności, które chciał dołączyć do AQP, ale zabrakło na to czasu. Zaczął zatem wstawiać nowe funkcje ulepszające AQP w Azure SQL Server Database, planując włączenie ich do wydania SQL Server 2019. Co więcej, słowo *adaptacyjny* nie odzwierciedlało naprawdę tej pracy, którą zespół realizował. Procesor zapytań SQL Server od lat był całkiem sprytny – używał wyrafinowanego zbioru algorytmów bazujących na kosztach do podejmowania decyzji o planie wykonania. Jednak zespół chciał więcej; dążył do tego, aby QP eksponował więcej inteligencji. Stąd nazwa *Intelligent Query Processing*.

Rysunek 2-1 pokazuje rodzinę funkcjonalności QP, które zawiera zarówno wydanie SQL Server 2017, jak i 2019.



Rysunek 2-1 *Rodzina funkcjonalności Intelligent Query Processing*

Przyjrzyjmy się każdej z nowych funkcjonalności, które zostały wyróżnione szarym kolorem na rysunku 2-1, z przykładami działania. W trakcie lektury tego podrozdziału trzeba pamiętać, że zbudowaliśmy je tak, **aby użytkownik nie musiał o nich wiedzieć**. Jeśli dostatecznie dobrze wykonaliśmy swoją pracę, Intelligent Query Processing jest „po prostu" procesorem zapytań, zaś programista aplikacji, administrator czy analityk danych może go używać jak dawniej, tyle że jako silnika bardziej elastycznego, inteligentnego i dostosowującego się do obciążenia. Wszystkie możliwości mechanizmu Adaptive Query Processing można znaleźć w nowej dokumentacji IQP pod adresem `https://docs.microsoft.com/en-us/sql/relational-databases/performance/intelligent-query-processing`.

Uwaga We wszystkich scenariuszach z wyjątkiem *Approximate Count Distinct* można włączyć funkcjonalność Intelligent Query Processing, zmieniając poziom zgodności bazy danych na 150. Approximate Count Distinct jest funkcją T-SQL nową w SQL Server 2019 i nie wymaga poziomu zgodności bazy danych równego 150.
