**Omówienie podstaw bazy danych**

Jak widać, nauka o danych kręci się wokół lepszego zrozumienia danych. Dlatego będziesz pracować z bazami danych, aby uzyskać dostęp do danych potrzebnych do zadawania interesujących pytań. Istnieje wiele różnych typów baz danych. Ponadto istnieje wiele terminologii używanej specjalnie dla baz danych. Musisz znać podstawowe pojęcia i terminy używane w świecie baz danych oraz sposób organizacji różnych baz danych.

**Tworzenie połączeń z relacyjnymi bazami danych**

Analitycy danych będą pracować z danymi w wielu różnych formach, w tym w starszych bazach danych lub starych arkuszach kalkulacyjnych. Mogą również pracować ze zdjęciami i filmami. Jako specjalista ds. danych powinieneś znać typowe sposoby przechowywania danych przez organizacje. Większość organizacji posiada szeroką gamę baz danych. Niektóre z nich są bardzo nowoczesne, inne mniej. Najlepszym sposobem na zrozumienie tych różnych technologii jest rozpoczęcie od początku. Nawet najnowocześniejsze bazy danych są często budowane w oparciu o technologię, która ma ponad 50 lat. Współczesne bazy danych naprawdę zaczęły się wraz z misją kosmiczną Apollo pod koniec lat 60. XX wieku. Rakiety, które miały polecieć na Księżyc, wymagały milionów części, a NASA współpracowała z IBM, aby stworzyć system zarządzania informacją, czyli IMS, w celu uporządkowania tych danych. Agencja kosmiczna miała wczesne manifesty, które bardzo przypominały nowoczesny arkusz kalkulacyjny. Były to pliki komputerowe z ciągami kolumn i długimi listami wierszy. Jak możesz sobie wyobrazić, zarządzanie tabelą z milionem wierszy może być trudne na małym terminalu czarno-białym. Mniej więcej w tym samym czasie agencja kosmiczna wykorzystała pierwsze relacyjne bazy danych. Bazy te podzieliły dane na grupy tabel. Każda z tych tabel nadal wyglądała jak arkusz kalkulacyjny, ale przedstawiała mniejszy fragment danych. Następnie stworzyli relacje między tymi tabelami. Zamiast jednej długiej listy obejmującej milion części, mogli stworzyć 50 tabel po 20 000 części każda. Dlatego nazywamy je relacyjnymi bazami danych. Baza danych jest oparta na grupach tabel, które są ze sobą powiązane. Nawet pierwsi inżynierowie baz danych mieli trudności ze stworzeniem wydajnego sposobu grupowania tabel bazy danych. Stworzyli mapy, aby pokazać, jak tabele są ze sobą powiązane. Nazywają te mapy schematami. Schemat jest tym, co sprawia, że relacyjna baza danych jest łatwa w użyciu lub koszmarna w zarządzaniu. Nawet przy tych wczesnych bazach danych widać, jak inżynierowie mogli mieć problemy z tworzeniem schematów. Czy powinni tworzyć tabele wokół największych części? Może zrobić tabelę na same stery strumieniowe, a potem kolejną tabelę na zbiornik paliwa? Problem polega na tym, że jeśli zmienisz projekt rakiety, to musisz również zmienić projekt bazy danych. Może mógłbyś stworzyć tabele na podstawie producenta części. Problem polega na tym, że możesz mieć producenta, który produkuje tysiące części i innego producenta, który produkuje tylko kilkadziesiąt. To nadal jest wyzwaniem dzisiaj. Relacyjne bazy danych wymagają dużej ilości wstępnego projektu. Musisz dużo wiedzieć o tym, jak będą wyglądać Twoje dane, zanim zaczniesz je zbierać. Jeśli się mylisz, przeprojektowanie wymaga dużo wysiłku Twojej bazy danych. IBM później skomercjalizował IMS, który stworzyli dla NASA. W połowie lat 70. opracowali Structured Query Language (SQL), aby pomóc swoim klientom pobierać dane z systemu. Ten język jest nadal bardzo popularny. SQL to elegancki język, który może pobierać dane z kilku różnych tabel relacyjnych. Ponownie łączy wszystkie różne tabele i przedstawia dane tak, jakby wszystkie były przechowywane w jednym dużym arkuszu. Ta wirtualna tabela jest powszechnie nazywana „widokiem”. Na przestrzeni lat do relacyjnych baz danych dodano wiele funkcji. Dały one początek systemowi zarządzania relacyjnymi bazami danych (RDBMS). Firmy takie jak IBM, Microsoft i Oracle nadal wspierają i rozwijają systemy zarządzania relacyjnymi bazami danych.

**Uwaga** : Innym terminem dotyczącym relacyjnej bazy danych, który możesz usłyszeć, jest CRUD , co oznacza tworzenie, odczytywanie, aktualizowanie i usuwanie. Opisuje wszystkie funkcje RDBMS. Czasami ludzie umieszczają S przed „wyszukaj" i używają akronimu SCRUD

**Wprowadzanie danych do magazynów za pomocą ETL**

Terminy i pojęcia omówione w tej sekcji są używane przez zespoły analizy danych. Staraj się nie przytłaczać językiem. Jeśli rozumiesz terminy i wyzwania, masz większe szanse na szybkie uzyskanie potrzebnych danych. Wiele koncepcji nauki o danych opiera się na wcześniejszych pracach z relacyjnymi bazami danych. Firmy od dziesięcioleci zbierają i próbują analizować dane. Nawet dzisiaj RDBMS jest nadal podstawą korporacyjnych baz danych i musisz rozumieć terminy RDBMS dotyczące projektów z zakresu nauki o danych. Jednym z miejsc, w których prawdopodobnie spotkasz się z terminami dotyczącymi RDBMS, jest praca z hurtownią danych przedsiębiorstwa (EDW) . EDW to specjalny rodzaj relacyjnej bazy danych, która koncentruje się na analizie danych. Tradycyjne bazy danych są zoptymalizowane pod kątem przetwarzania transakcji online (OLTP). EDW jest używany do przetwarzania analitycznego online (OLAP). Pomyśl o tym w ten sposób: typowa baza danych koncentruje się na pracy z danymi w czasie rzeczywistym, podczas gdy EDW koncentruje się na analizie tego, co już się wydarzyło. Wyobraźmy sobie, że masz witrynę internetową, która sprzedaje buty do biegania. Zatrudniłeś inżyniera do stworzenia bazy danych. Stworzył dziesiątki różnych tabel i relacji. Jest tabela z adresami klientów, tabela z butami, tabela z opcjami wysyłki i tak dalej. Serwer WWW używa instrukcji SQL do zbierania danych. Gdy klient kupuje parę butów, jego dane adresowe są kojarzone z butem, serwer sieciowy udostępnia klientowi opcje wysyłki i para butów jest wysyłana. Chcesz, aby ta baza danych była szybka i wydajna oraz skupiała się na szybkich zwrotach. Ta baza danych to to, co robi Twój klient w czasie rzeczywistym. To jest baza danych OLTP. Poprosisz również inżyniera bazy danych o utworzenie skryptu, który codziennie przesyła dane do magazynu. Twoja hurtownia danych jest zoptymalizowana pod kątem przetwarzania analitycznego. Jest to baza danych OLAP skoncentrowana na tworzeniu raportów. Analityk danych tworzy raport, aby sprawdzić, czy istnieje jakikolwiek związek między adresem klienta a rodzajem butów, które kupuje. Przekonasz się, że ludzie w cieplejszych obszarach chętniej kupują buty w jasnych kolorach. Wykorzystujesz te informacje, aby zmienić swoją stronę internetową, aby klienci z cieplejszych klimatów widzieli jaśniejsze buty na górze strony. Teraz powiedz, że Twoja strona internetowa odniosła duży sukces i została kupiona przez firmę sprzedającą wszystkie rodzaje odzieży sportowej. Ta firma ma jeden magazyn dla wszystkich swoich witryn internetowych i chce połączyć dane z Twojej witryny internetowej ze wszystkimi innymi witrynami internetowymi. W tym momencie firma zrobi coś, co nazywa się ETL , co oznacza wyodrębnianie, przekształcanie i ładowanie . Pobierają dane z twojej strony internetowej, a następnie ładują je do swojego EDW. Kiedy wyodrębnią Twoje dane, spróbują zrobić to w jakimś standardowym formacie, aby móc przekształcić dane w coś, co dobrze współpracuje z ich hurtownią danych. Tabele z ich magazynu mogą mieć inny schemat. Na przykład hurtownia danych może zawierać informacje o wysyłce w tabeli Customer, podczas gdy baza danych zawiera informacje o wysyłce we własnej tabeli. Dane muszą zostać przekształcone, aby trafiły do EDW. Analityk Danych najprawdopodobniej spędzi większość czasu na szorowaniu i łączeniu danych tak, aby się zmieściły, a następnie w końcu załaduje przetworzone dane do hurtowni. Poprzedni scenariusz nie jest jedynym, w którym może być konieczne wykonanie ETL. Niektóre firmy mogą mieć hurtownię danych oddzieloną od klastra Hadoop, w takim przypadku będą musiały uruchomić ETL na danych hurtowni, aby przenieść je do klastra Hadoop. W takim przypadku analityk danych musi przekształcić dane, aby można było ich używać w klastrze. Wiele organizacji często postrzega Hadoop jako zamiennik kosztownych hurtowni danych. Chcą zaoszczędzić pieniądze, przechowując dane na niedrogim sprzęcie zamiast na kosztownym urządzeniu magazynowym. W takim przypadku firmy mogą przepisać swoje procedury ETL, aby mogły załadować dane do klastra Hadoop, a następnie wycofać lub zamknąć magazyn.

**Odejście od przeszłości z NoSQL**

Często zespół zajmujący się badaniem danych potrzebuje bardziej elastycznego sposobu przechowywania danych. Pamiętaj, że relacyjne bazy danych opierają się na schemacie. Musisz dużo wiedzieć o swoich danych, zanim będziesz mógł je umieścić w bazie danych, co oznacza, że musisz planować z wyprzedzeniem. Musisz wiedzieć, jaki typ danych pojawia się w polach bazy danych (tekst, wideo, audio lub inne), zorganizować te pola w tabele, a następnie utworzyć relacje między tabelami. Baza danych wymaga ustalonej struktury, dzięki czemu możesz tworzyć, czytać, aktualizować i usuwać swoje rekordy. W przypadku niektórych bardzo dużych baz danych to obciążenie może spowodować zablokowanie serwerów. Wróćmy do Twojej witryny internetowej z butami do biegania. Klient znajduje parę butów i przechodzi na stronę kasy. W tym momencie strona internetowa łączy parę kupowanych butów z adresem klienta. Ta strona kasy wymaga dostępu do czterech różnych tabel bazy danych:

• Tabela na buty

• Tabela klienta

• Tabela adresów

• Tabela wysyłkowa

To dużo pracy dla bazy danych. Im ciężej działa twoja baza danych, tym wolniej twoja strona internetowa. Jak to przyspieszyć? Potrzebujesz kupić większy serwer, podzielić stoły na kilka serwerów lub mieć kilka serwerów, które synchronizują się w sieci? W przypadku naprawdę dużych witryn internetowych te opcje mogą wydawać się nienaturalne. Teraz wyobraź sobie bazę danych, która przechowuje wszystko na stronie kasy jako jedną transakcję. Transakcja bazy danych to jeden fragment pracy, który musi wykonać wszystko albo nic. Tworzony jest rekord dla pary butów, klienta, jego adresu i wysyłki – wszystko w jednym ujęciu. Co teraz, jeśli dane nie są podzielone na tabele i nie trzeba sprawdzać relacji? Informacje są po prostu wrzucane i gotowe. To jest idea NoSQL. NoSQL został po raz pierwszy użyty jako hashtag na Twitterze dla programistów, którzy chcieli wyjść poza relacyjne bazy danych. W rzeczywistości nie jest to atak przeciwko SQL. W rzeczywistości NoSQL w ogóle nie ma wiele wspólnego z SQL. Chodzi bardziej o ograniczenia relacyjnych baz danych. Ogólnie rzecz biorąc, baza danych NoSQL powinna być nierelacyjna, pozbawiona schematów, przyjazna dla klastrów i, miejmy nadzieję, open source. Wszystkie te cechy powinny spodobać się zespołowi zajmującemu się analizą danych. Baza danych, która nie jest relacyjna, jest zazwyczaj łatwiejsza do zmiany i prostsza w użyciu. Nie musi istnieć rozbieżność między wyglądem Twojej aplikacji internetowej a sposobem przechowywania danych. Nie będziesz też musiał przechodzić przez brzydki proces tworzenia i dzielenia tabel, które już istnieją. Jest to powszechnie określane jako normalizowanie bazy danych. Bez schematu nie musisz się martwić, że wiesz wszystko z góry. Wróćmy do strony internetowej z butami do biegania. Kupiła go większa firma. Ta firma chce dodać Twoich klientów do swojego programu dla częstych nabywców. W przypadku relacyjnej bazy danych jest to poważne wyzwanie architektoniczne. Czy w tabeli klientów powinien znajdować się numer częstego kupującego? Czy musisz utworzyć zupełnie nową tabelę, aby przechowywać tylko numery często kupujących? Czy klient może mieć więcej niż jeden numer kupującego? Czy dwóch klientów może dzielić ten sam numer? Zanim klienci będą mogli zostać dodani do programu dla częstych nabywców, wszystkie te pytania muszą zostać rozwiązane. Musisz przerobić bazę danych i dowiedzieć się, jak poprawić brakujące dane. Bez schematu nowe pola stają się niemal trywialne. Po prostu przechowujesz to jako jedną transakcję. Jeśli klient ma częsty numer kupującego, pojawia się on w transakcji. Jeśli klient go nie posiada, pole nie istnieje. Wreszcie baza danych NoSQL powinna być przyjazna dla klastrów. Powinieneś być w stanie przechowywać dane na kilkuset lub nawet tysiącach serwerów bazodanowych. W NoSQL rekord zapisany w transakcji nazywany jest agregatem . Te agregaty zawierają wszystkie dane: informacje o

butach, kliencie, adresie i wysyłce. Te agregaty są łatwiejsze do synchronizowania na wielu serwerach baz danych. Większość serwerów pracuje w klastrach. Dzięki temu mogą synchronizować się między sobą, a następnie wysyłać aktualizacje do innych klastrów. Kiedy pracujesz w zespole zajmującym się analizą danych, prawie na pewno natkniesz się na NoSQL. Dla wielu organizacji jest to preferowany sposób radzenia sobie z dużymi zestawami danych. Ze względu na prostszą konstrukcję programiści mogą również znacznie łatwiej tworzyć aplikacje internetowe, które mogą szybko rozrosnąć się do skali przedsiębiorstwa.

**Uwaga**: słowo „klaster" powinno brzmieć znajomo. Jest to ten sam sposób, w jaki Hadoop działa ze swoimi zestawami danych. W rzeczywistości większość Hadoop jest oparta na HBase, która jest bazą danych NoSQL o otwartym kodzie źródłowym.

**Problem dużych zbiorów danych**

Jak wspomniano wcześniej, big data i data science są ze sobą tak powiązane, że wiele organizacji postrzega je jako to samo. Pamiętaj, że nauka o danych wykorzystuje metodę naukową do zadawania interesujących pytań. Nie oznacza to, że potrzebujesz dużo danych, aby zadać te pytania. Big data zapewnia bogate nowe źródło danych, które pozwala zadawać pytania, na które nie można odpowiedzieć przy użyciu mniejszego zestawu danych. Big data nie jest tak naprawdę rzeczownikiem. W oryginalnym artykule NASA został opisany jako „problem z dużymi danymi". Możesz to przeczytać na dwa sposoby: jest to problem „dużych danych" lub duży „problem z danymi". Jeśli przeczytasz cały artykuł, wydaje się, że kładą nacisk na problem. Nie chodzi o „duże zbiory danych". Chodzi o problem, co zrobić z tymi wszystkimi nowymi danymi. Jest to również poruszone dekadę później w raporcie McKinseya. W raporcie autorzy odnoszą się do big data jako danych, które przekraczają możliwości powszechnie używanego sprzętu i oprogramowania. Dlaczego więc ważne jest, aby myśleć o big data jako o problemie, a nie o rzeczowniku? Cóż, to dlatego, że wiele firm, które rozpoczynają projekty big data, w rzeczywistości nie ma big data. Może się wydawać, że jest duży, ponieważ jest go dużo. Wydaje się to również problemem, ponieważ przechowywanie i gromadzenie to prawdziwe wyzwanie. Ale nie jest to „problem z dużymi danymi". Jednym ze sposobów określenia, czy masz problem z dużymi danymi, jest sprawdzenie, czy Twoje dane należą do czterech kategorii. Możesz zapamiętać te kategorie jako cztery V. Zadaj sobie te pytania:

• Czy mam dużą ilość danych?

• Czy mam szeroką gamę danych?

• Czy dane napływają z dużą prędkością?

• Czy zbierane przeze mnie dane mają wiarygodność?

Aby być big data, musi mieć wszystkie cztery z tych atrybutów. Pytanie dotyczące głośności jest zwykle dość proste. Jeśli codziennie zbierasz petabajty danych, prawdopodobnie masz wystarczającą ilość danych. Oczywiście nie zawsze może to stanowić problem. Być może w niedalekiej przyszłości eksabajt zostanie uznany za wystarczająco wysoki wolumen, aby stanowić problem. Powinna istnieć szeroka gama informacji. Może zawierać tekst, wideo, dźwięk i obrazy. Jeśli chodzi o szybkość, pomyśl o giełdzie nowojorskiej. Każdego dnia obsługują miliardy transakcji. Mają dużą ilość danych napływających z dużą prędkością. Ceny akcji napływają i zmieniają się w milisekundach. Jednak to wszystko ten sam rodzaj danych. Zwykle jest to tylko symbol giełdowy i tekst zawierający głównie cenę. Zbierają dane o transakcjach, a nie zdjęcia, dźwięki czy wiadomości. Więc nie mają problemu z dużymi danymi. Z pewnością zbierają dużo danych, ale technologia, którą mają, powinna być więcej niż zdolna do sprostania temu wyzwaniu. Na koniec pomyśl o prawdziwości danych. Wyobraź sobie, że stworzyłeś

bazę danych, która zebrał wszystkie tweety i posty na Facebooku dotyczące Twojej witryny. Zbierasz filmy, zdjęcia i tekst. Każdego dnia do klastra przesyłanych jest kilka petabajtów danych. Tworzysz raporty, aby sprawdzić, czy klienci są pozytywnie nastawieni do Twojego produktu. Po przejrzeniu danych zdajesz sobie sprawę, że nie ma tam pytania, które określiłoby nastrój klienta. Cały ten wysiłek poświęcono na zbieranie bezużytecznych danych, ponieważ te dane nie dostarczają żadnych potrzebnych informacji.

Aby przedstawić interesujący przykład problemu z dużą ilością danych, pomyśl o wyzwaniu związanym z autonomicznymi samochodami. Jaki rodzaj danych musiałbyś zbierać? Musiałbyś zbierać ogromne ilości wideo, dźwięków, raportów o ruchu drogowym i informacji o lokalizacji GPS - wszystko to wpływałoby do bazy danych w czasie rzeczywistym i z dużą prędkością. Wtedy samochód musiałby dowiedzieć się, jakie dane mają najwyższą prawdziwość. Czy ta osoba na bocznej drodze krzyczy z powodu meczu sportowego, czy krzyczy, ponieważ ktoś stoi na drodze? Kierowca ma sekundy, żeby to rozgryźć. Samochód z dużymi danymi musiałby natychmiast przetworzyć obraz, dźwięk i współrzędne ruchu, a następnie zdecydować, czy ma się zatrzymać, czy po prostu zignorować dźwięk. To prawdziwy problem z dużymi danymi.

**Wskazówka** : spróbuj zapamiętać różnicę między big data a nauką o danych. Big data pozwoli Ci zadawać ciekawsze pytania. Nie oznacza to, że wszystkie interesujące pytania wymagają dużych zbiorów danych. Skoncentruj się na nauce. W ten sposób, niezależnie od posiadanych danych, zawsze będziesz mógł zadać najlepsze pytania.

**Podsumowanie**

Dowiedziałeś się, że ponieważ nauka o danych obraca się wokół interesujących danych, często musisz pracować z kilkoma rodzajami baz danych. Zapoznałeś się z terminologią używaną specjalnie dla baz danych, a także z podstawowymi pojęciami i terminami dotyczącymi technologii. Widziałeś również, jak zorganizowane są bazy danych. Wkrótce dowiesz się, jak rozpoznawać różne typy danych