

Bazy Danych NoSQL

Robert A. Kłopotek

r.klopotek@uksw.edu.pl

Wydział Matematyczno-Przyrodniczy. Szkoła Nauk Ścisłych, UKSW

Wady i zalety relacyjnych baz danych

► Zalety

- Możliwość trwałego składowania danych
- Relacyjne bazy danych oferują większą elastyczność
- Współbieżność
- Obsługa błędów
- System transakcji
- Język SQL

► Wady

- W bazach transakcyjnych ACID przeszkadza w pracy w klastrze.
- Model danych w relacyjnych bazach danych jest mniej elastyczny, niż w przypadku rozwiązania z rodziny NoSQL.

Bazy danych wielowymiarowe

- ▶ Wielowymiarowa baza danych (multidimensional database - MDB) to typ bazy danych, zoptymalizowany pod kątem hurtowni danych i aplikacji do przetwarzania danych on-line (online analytical processing - OLAP)
- ▶ Wielowymiarowe bazy danych są często tworzone przy użyciu danych wejściowych z istniejących relacyjnych baz danych.
- ▶ Aplikacja OLAP, która uzyskuje dostęp do danych z wielowymiarowej bazy danych, est znana jako aplikacja MOLAP (multidimensional OLAP).
- ▶ System zarządzania wielowymiarową bazą danych (MDDDBMS) - daje możliwość szybkiego przetwarzania danych online w bazie danych

Bazy danych wielowymiarowe

- ▶ Konceptyjnie wielowymiarowa baza danych wykorzystuje ideę kostki danych do reprezentowania wymiarów danych dostępnych dla użytkownika.
- ▶ Na przykład "sprzedaż" można było zapoznać w wymiarze modelu produktu, geografii, czasu lub dodatkowego wymiaru. W tym przypadku "sprzedaż" jest znany jako atrybut środka kostki danych, a pozostałe wymiary są postrzegane jako atrybuty funkcji.
- ▶ Ponadto twórca bazy danych może zdefiniować hierarchie i poziomy w wymiarze (na przykład poziomy stanu i miasta w ramach hierarchii regionalnej).
- ▶ Mamy tu odczynienia z hybrydą bazy danych jako kostka OLAP (zawierająca szybki dostęp do agregatów) oraz „powolna” baz danych relacyjna

Bazy danych NoSQL

- ▶ NoSQL i Not Only SQL opisują podejście do projektowania baz danych, które implementuje magazyn danych kluczy i wartości, magazyn dokumentów, magazyn kolumn lub magazyny z grafami.
- ▶ Jest to alternatywa dla bazy danych SQL (Structured Query Language), która rozpoczyna się w latach osiemdziesiątych.
- ▶ NoSQL kontrastuje z bazami, które stosują się do metod relacyjnych SQL, gdzie dane są umieszczane w tabelach i schematach danych są starannie zaprojektowane przed zbudowaniem bazy danych.
- ▶ Bazy danych NoSQL szczególnie dotyczą dużych zbiorów danych rozproszonych.

NoSQL vs. RDBMS

- ▶ NoSQL bardziej powszechnie odnosi się do baz danych zbudowanych na początku 2000 roku w celu tworzenia na dużą skalę baz danych w aplikacjach chmurowych i internetowych. W tych zastosowaniach wymagania dotyczące wydajności i skalowalności przewyższały potrzebę natychmiastowej, sztywnej spójności danych, którą RDBMS dostarczył do aplikacji korporacyjnych.
- ▶ Systemy NoSQL nie musiały przestrzegać ustalonego schematu relacyjnego. Wielkoskalowe organizacje internetowe, takie jak Google i Amazon, korzystały z baz danych NoSQL, koncentrując się na wąskich celach operacyjnych i stosowaniu relacyjnych baz danych jako elementów uzupełniających, w których konieczne jest zachowanie wysokiej jakości spójności danych.
- ▶ Wczesne bazy danych NoSQL dla aplikacji sieciowych i chmurowych koncentrowały się na bardzo specyficznych cechach zarządzania danymi. Możliwość przetwarzania bardzo dużej ilości danych i szybka dystrybucja tych danych w klastrach obliczeniowych była pożądaną cechą w projektowaniu stron internetowych i chmur.

Rodzina baz danych NoSQL

All in the NoSQL Family

NoSQL databases are geared toward managing large sets of varied and frequently updated data, often in distributed systems or the cloud. They avoid the rigid schemas associated with relational databases. But the architectures themselves vary and are separated into four primary classifications, although types are blending over time.



Document databases

Store data elements in document-like structures that encode information in formats such as JSON.



Common uses include content management and monitoring Web and mobile applications.



EXAMPLES:

Couchbase Server, CouchDB, MarkLogic, MongoDB



Graph databases

Emphasize connections between data elements, storing related “nodes” in graphs to accelerate querying.



Common uses include recommendation engines and geospatial applications.



EXAMPLES:

Allegrograph, IBM Graph, Neo4j



Key-value databases

Use a simple data model that pairs a unique key and its associated value in storing data elements.



Common uses include storing clickstream data and application logs.



EXAMPLES:

Aerospike, DynamoDB, Redis, Riak



Wide column stores

Also called table-style databases—store data across tables that can have very large numbers of columns.



Common uses include Internet search and other large-scale Web applications.



EXAMPLES:

Accumulo, Cassandra, HBase, Hypertable, SimpleDB

Bazy danych typu kucz-wartość

- ▶ Jest to najprostsza implementacja NoSQL.
- ▶ W tym wypadku opieramy się na mapie, która pozwala dodawać i odczytywać wartości poprzez odwołanie się do klucza.
- ▶ Przykładem bazy danych klucz wartość jest np. Riak, bądź Redis.
- ▶ Przykładowy kod, który można wykorzystać w kontrolerze, jeżeli korzystamy z REDIS:

```
$redis = Redis::connection();  
$redis->set('company', 'Grupa Tense');  
$company= $redis->get('company');  
echo $company;
```

Bazy danych zorientowane na dokumenty

- ▶ Inaczej „Document-oriented database”
- ▶ Dokumenty to samoopisujące się, hierarchiczne struktury drzewiaste, które mogą składać się z map, kolekcji i wartości.
- ▶ Przechowywane dokumenty są do siebie podobne, ale nie muszą być dokładnie takie same.
- ▶ Bazy tego typu przechowują dane w postaci klucz-kłucz wartość.
- ▶ Przykład struktury danych:
{ imie: "Jan",
nazwisko: "Nowak",
nr_indeksu: 1174145,
oceny: [5, 3, 4.5, 3.5, 4],
dzienny: true}
- ▶ Przykładem bazy danych kluczem wartość jest np. MongoDB, RavenDB.

Bazy danych pełnotekstowe

- ▶ Inaczej „full text database” lub „complete text database”
- ▶ To kompilacja dokumentów lub innych informacji w formie bazy danych, w której dostępny jest pełny tekst każdego referencyjnego dokumentu do przeglądania, drukowania lub pobierania online.
- ▶ Oprócz dokumentów tekstowych obrazy są często dołączane, na przykład wykresy, mapy, zdjęcia i diagramy.
- ▶ Bazę danych pełnotekstowych można wyszukiwać słowami kluczowymi, frazą lub obydwojako.
- ▶ Jeżeli pozycja w bazie danych pełnotekstowych jest oglądana, może pojawić się w formacie ASCII (jako plik tekstowy z rozszerzeniem .txt), jako przetworzony plik Word (wymagający program takich jak Microsoft Word), lub jako plik PDF. Gdy dokument jest wyświetlany jako plik PDF, jest to zwykle zeskanowany (hardcopy) oryginalnego artykułu, rozdziału lub książki.

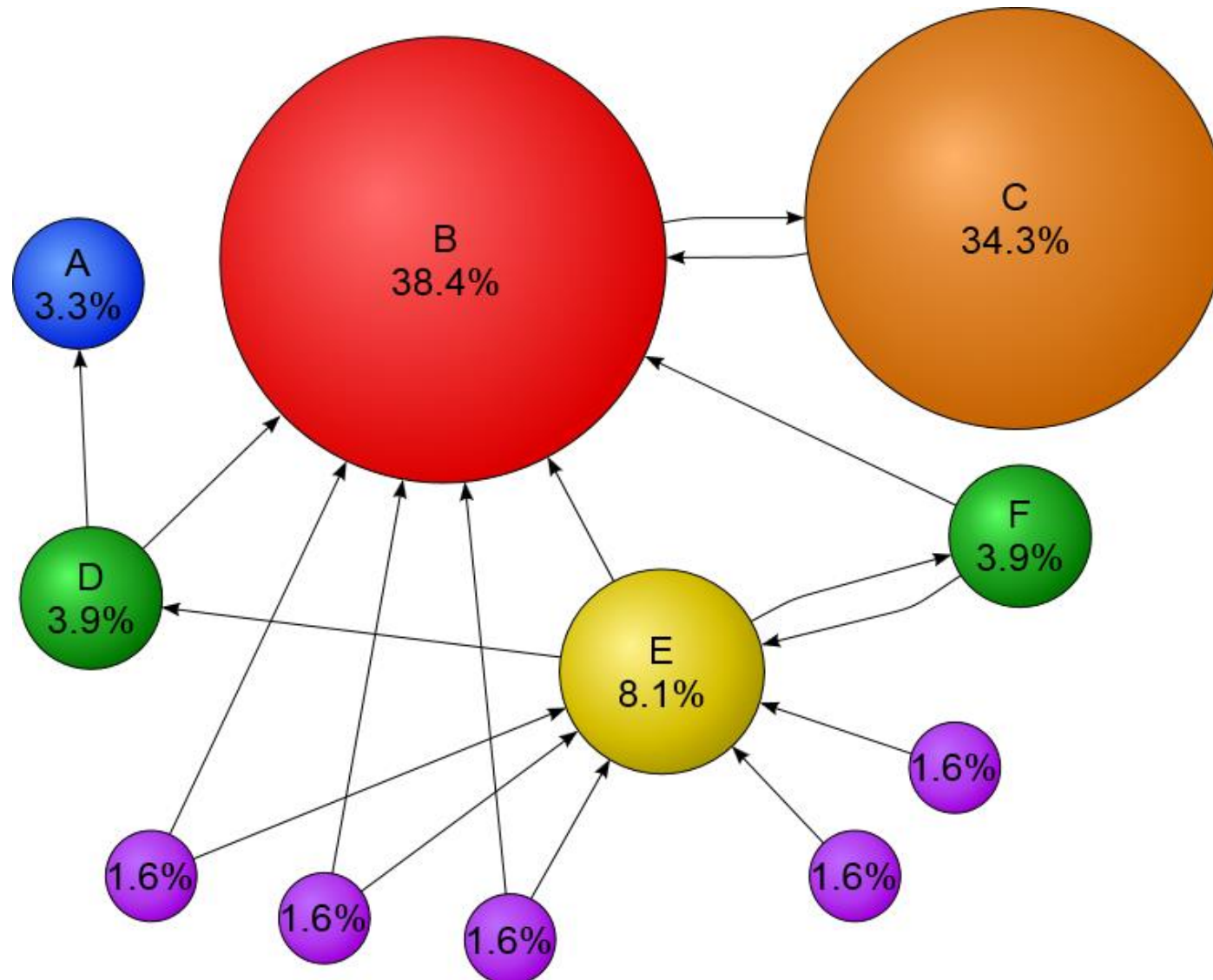
Hiperteksty

- ▶ SJP: „hipertekst «sposób organizacji informacji w tekście komputerowym, polegający na zastosowaniu wyróżnionych odsyłaczy, które automatycznie przenoszą użytkownika do innych informacji; też: tekst zorganizowany w ten sposób»”
- ▶ Oznacza to, że nie ma z góry zdefiniowanej kolejności czytania leksji, a nawigacja między nimi zależy wyłącznie od użytkownika.
- ▶ W hipertekście konwencjonalnym zbiór węzłów jest zamknięty, zaczepienia powiązań umieszczone są na stałe w węzłach w czasie tworzenia systemu, a urządzenia wyszukiwawcze działają w oparciu o ustalone indeksy oraz ustalone sieci powiązań węzłów.
- ▶ W hipertekstach drugiego typu (bazodanowych) oddziela się bazę węzłów od struktury powiązań. Takie niezależne struktury są łatwe do modyfikacji, a także pozwalają na tworzenie różnych sieci powiązań dla jednej bazy węzłów. Urządzenia wyszukiwawcze muszą wtedy działać na modyfikowalnych zbiorach danych.

Hipertekstowe bazy danych

- ▶ Struktura tradycyjnej - atrybutowej (!) - bazy danych reprezentowana jest w jej schemacie, a wszystkie zapytania odwołują się do tej sztywnej struktury: nazw tabel, nazw atrybutów i zakresów wartości atrybutów.
- ▶ W bazach danych hipertekstowych informacja przechowywana jest w węzłach różnego rodzaju mediów (np. teksty, dźwięki, filmy, obrazy) połączonych za pomocą „wiązań asocjacyjnych”
- ▶ Bazy takie oferują użytkownikom zarówno możliwość nawigacji, jak i korzystanie z „urządzeń wyszukiwawczych” - wyszukiwarek.
- ▶ Wyszukiwarka sieci grafowej, jaką stanowią dokumenty hipertekstowe (graf skierowany) jest zwykle oparta na algorytmie PageRank lub jego modyfikacji
- ▶ PageRank - metoda nadawania indeksowanym stronom internetowym określonej wartości liczbowej, oznaczającej ich jakość.
- ▶ Algorytm PageRank jest wykorzystywany przez popularną wyszukiwarkę internetową Google

PageRank - idea



Multimedialna baza danych

- ▶ Jest to hipertekstowa baza danych, która zawiera:
 - ▶ obiekty medialne (dyskretne: tekst i obraz, ciągłe: audio i wideo, oraz złożone) tworzące bazę multimedialną
 - ▶ system multimedialny (system wspierający wymianę informacji z użytkownikiem za pomocą kilku różnych mediów),
 - ▶ system zarządzania bazą, który wspiera i obsługuje dane medialne w zakresie ich
 - ▶ składowania, wyszukiwania i przetwarzania (czyli tak, jak SZRBD obsługują dane alfanumeryczne).
- ▶ Multimedialną zawartość da się „upchnąć” do popularnych relacyjnych bądź obiektowo, relacyjnych baz danych w postaci obiektów typu BLOB.
- ▶ Wtedy do każdego takiego obiektu dodaje się „nagłówek”, tj. pewien opis zawartości obiektu binarnego i wszelkie operacje wyszukiwania mogą uwzględniać wyłącznie zawartość tych nagłówków

Bazy danych oparte na XML

- ▶ Baza danych XML - programowy system trwałych struktur danych, który pozwala na zapisanie tych danych w formacie XML. Dane te mogą być potem pobierane, wysyłane i serializowane w dowolnym formacie.
- ▶ Dwie ważne klasy baz danych XML to:
 - ▶ Bazy danych umożliwiające przechowanie danych w formacie XML - są one tradycyjnymi bazami danych (np. relacyjnymi), które na wejściu i na wyjściu akceptują i generują dane w postaci XML. Przy czym ta konwersja jest wykonywana przez samą bazę danych, a nie przez dodatkowe oprogramowanie,
 - ▶ Natywne bazy danych XML - używają dokumentów XML jako podstawowe jednostki przechowywania.
- ▶ Powszechniejsze użycie technologii XML do przesyłania danych sprawia, że w przypadku istniejących baz danych, dane są pobierane z tych baz a potem konwertowane do dokumentów XML i na odwrót.

Natywne bazy danych XML

- ▶ Formalna definicja od Konsorcjum XML:DB stanowi że natywna baza danych XML:
 - ▶ Definiuje logiczny model dla dokumentu XML, a nie dla danych zawartych w tym dokumencie. Ponadto przechowuje i pobiera dokumenty zgodnie z danym modelem. Jako minimum przyjmuje się, że model musi zawierać elementy, atrybuty, elementy typu dowolny tekst (PCDATA) i kolejność w dokumencie.
 - ▶ Przechowuje dane w dokumentach XML jako podstawowych jednostkach przechowujących. Przykładowo, dla relacyjnych baz danych podstawową jednostką przechowującą jest wiersz tabeli,
 - ▶ Nie potrzebuje posiadać jakiegoś szczególnego fizycznego modelu przechowywania danych. Przykładowo, bazy danych XML mogą używać relacyjnych, hierarchicznych lub obiektowych struktur bazodanowych, a także zastrzeżonych formatów przechowujących, np. plików skompresowanych.

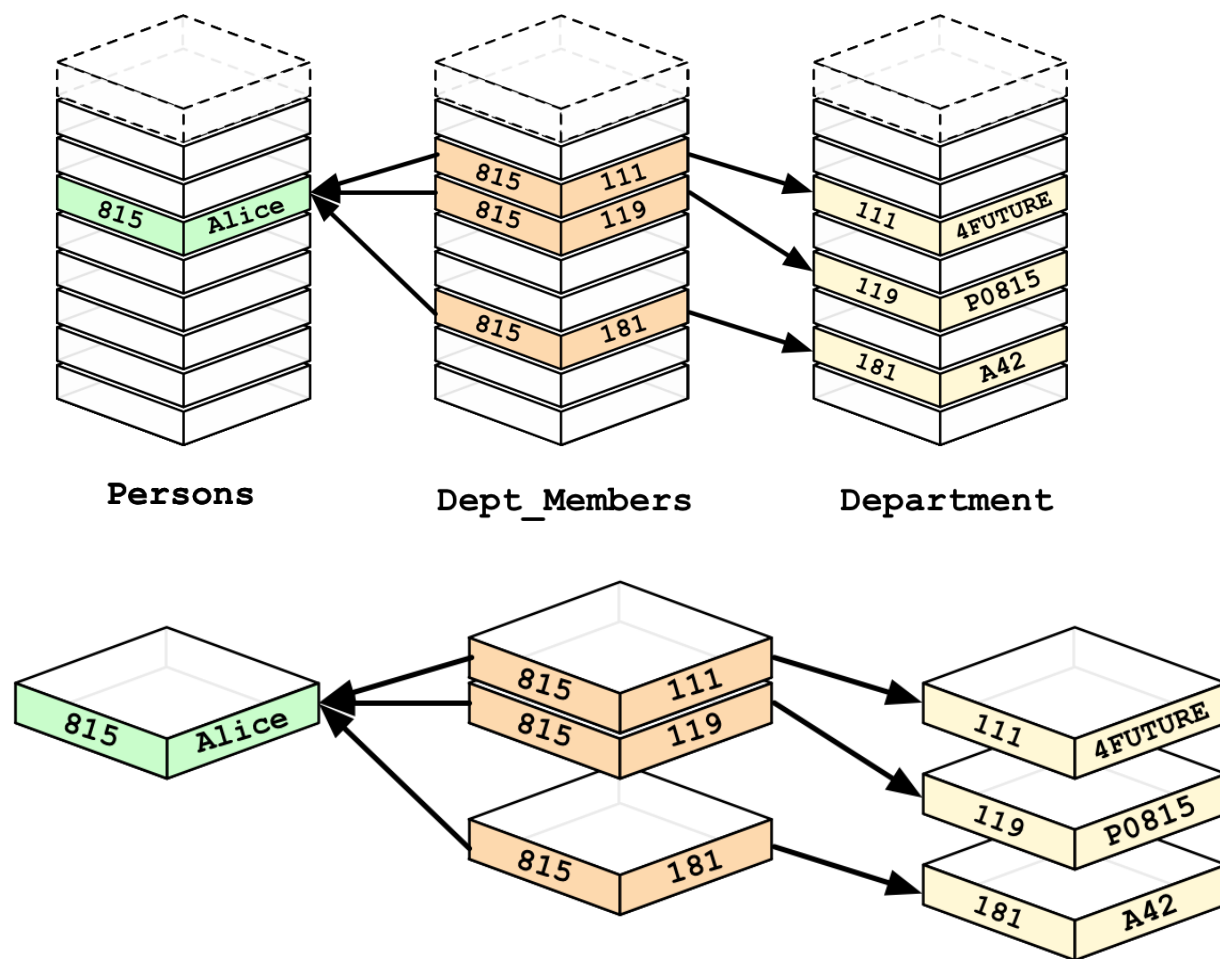
Sposoby komunikacji z baza danych XML

- ▶ Wszystkie bazy danych XML (od 2006 roku) obsługują co najmniej jedną formę składni zapytania.
- ▶ Prawie wszystkie z nich obsługują w stopniu co najmniej minimalnym XPath do przeprowadzania zapytań dla dokumentów lub kolekcji dokumentów.
- ▶ XPath jest prostym systemem wybierającym, pozwalającym użytkownikom na wyszczególnienie interesujących ich węzłów dokumentów XML spełniających określone przez nich kryteria.
- ▶ Oprócz XPatha wiele baz danych XML obsługuje XSLT jako metodę przetworzenia dokumentów lub wyników zapytań do tych baz.
- ▶ XSLT posiada deklaratywny język napisany w gramatyce XML. Jego celem jest definiowanie zbioru filtrów XPath, które posłużą do przetwarzania części lub całości dokumentów do innych formatów, np. czysty tekst, XML, HTML lub PDF.
- ▶ Nie wszystkie bazy danych XML obsługują XQuery do wykonywania zapytań. XQuery zawiera XPath jako metodę wyboru węzłów ale jednocześnie rozszerza ten XPath tak, aby posiadał pewne własności transformujące.

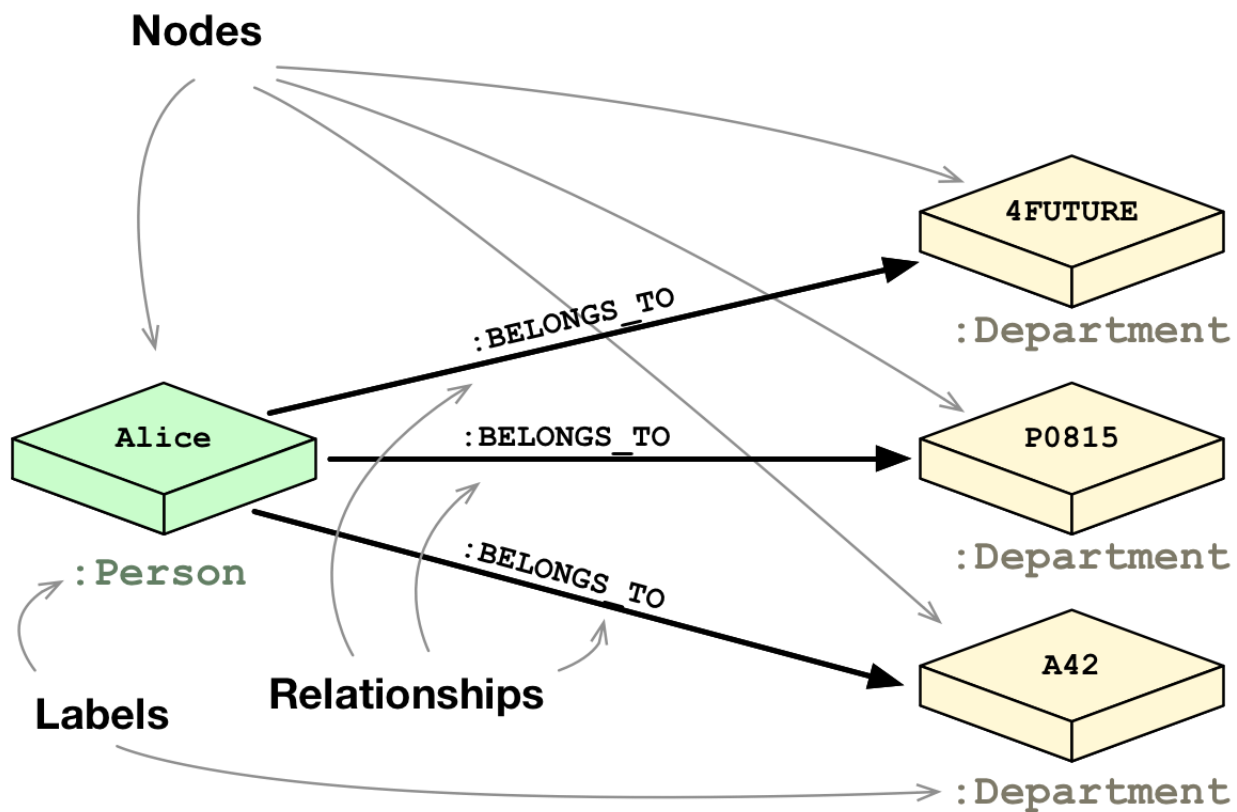
Bazy danych grafowe

- ▶ Bazy grafowe - baza danych, która wykorzystuje struktury grafów z węzłami.
- ▶ Można myśleć o węźle jako instancji obiektu, z krawędziami i własnościami do przedstawiania i przechowywania danych oraz do obsługi zapytań semantycznych.
- ▶ Implementacją takiego modelu grafowego jest np. struktura hierarchiczna w firmie.
- ▶ Rozwiązanie można oprzeć na takiej bazie danych jak AllegroGraph, IBM Graph, Neo4J, FlockDB.

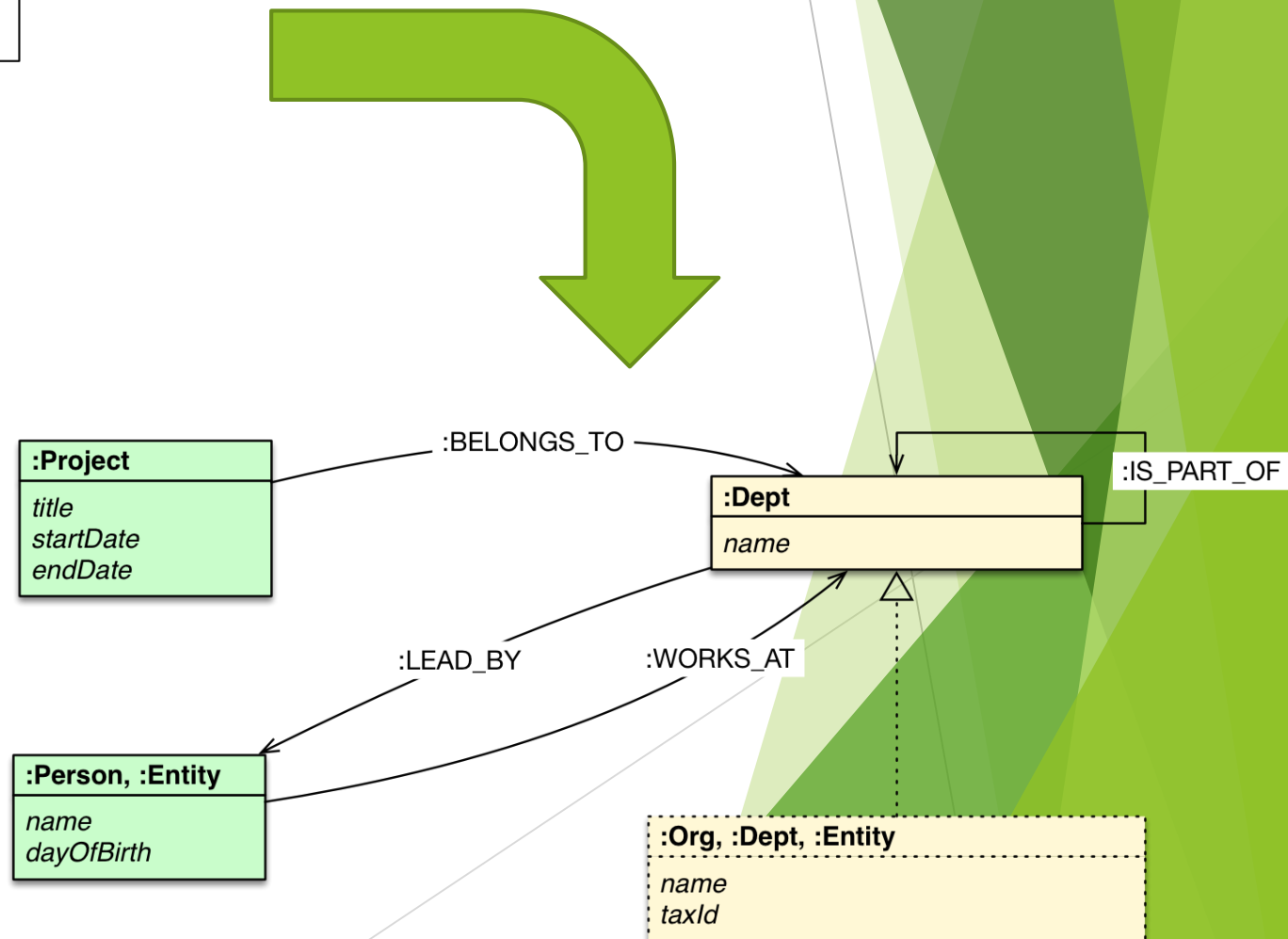
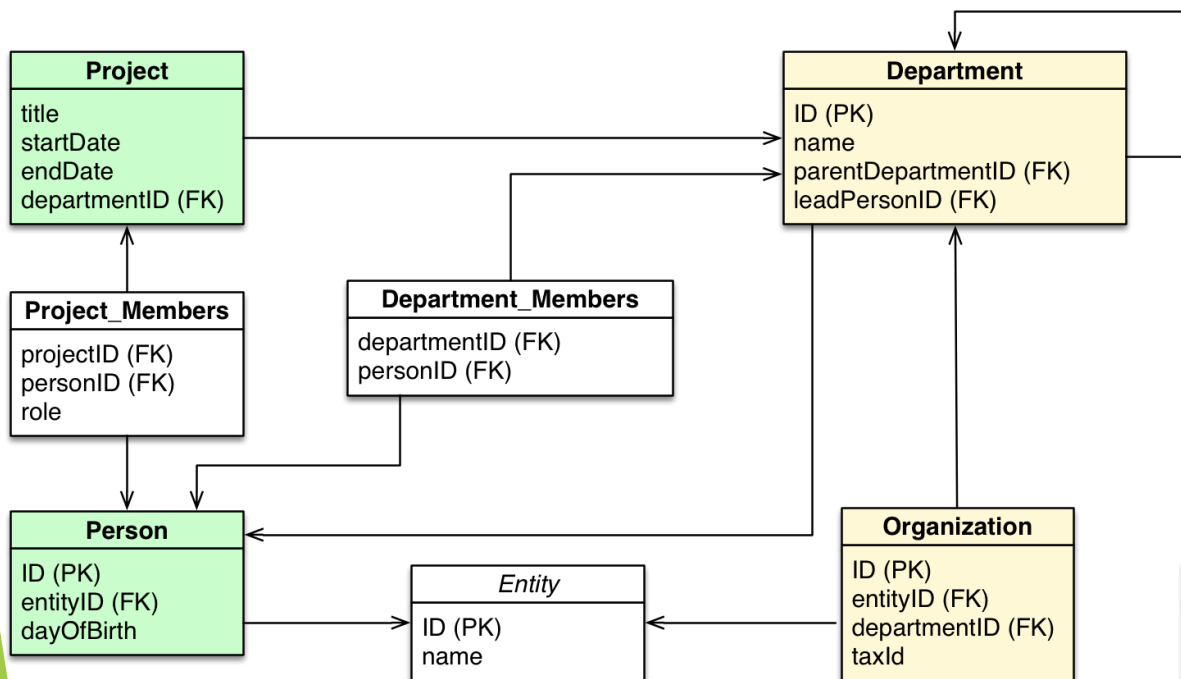
Reprezentacja informacji w grafowych bazach danych (1)



Reprezentacja informacji w grafowych bazach danych (2)



Model relacyjny vs model grafowy



Jak dokonać transformacji?

- ▶ Każda tabela encji jest reprezentowana przez etykietę na węzłach
- ▶ Każdy wiersz w tabeli encji jest węzłem
- ▶ Kolumny na tych tabelach stają się właściwościami węzła.
- ▶ Usunąć podstawowe klucze techniczne, zachować podstawowe klucze biznesowe
- ▶ Dodawanie unikalnych ograniczeń dla podstawowych kluczy biznesowych, dodawanie indeksów do częstych wyszukiwań
- ▶ Zastąpić klucze obce relacją (krawędzią), usuń je później
- ▶ Usuń dane z wartościami domyślnymi, nie musisz ich przechowywać
- ▶ Dane w tabelach, które są denormalizowane i duplikowane, mogłyby zostać wyjęte w oddzielne węzły, aby uzyskać bardziej przejrzysty model.
- ▶ Nazwy kolumn indeksowanych mogą wskazywać właściwość tablicy (np. email1, email2, email3)
- ▶ Połączenia tabel są przekształcane w relacje (krawędzie), kolumny na tych tabelach stają się właściwościami relacji

Przykład zapytań

- ▶ Składnia Cypher jest przejrzysta i skupiona na pojęciach domeny, ponieważ połączenia strukturalne do szukania lub tworzenia są wyrażane wizualnie (jako krawędzie grafu). Pozostałe klauzule, oprócz dopasowywania wzorca, są podobne do SQL.
- ▶ Zadanie: Wyświetl pracowników z działu IT
- ▶ SQL

```
SELECT name FROM Person
LEFT JOIN Person_Department
  ON Person.Id = Person_Department.PersonId
LEFT JOIN Department
  ON Department.Id = Person_Department.DepartmentId
WHERE Department.name = "IT Departmen"
```
- ▶ Cypher

```
MATCH (p:Person)<-[:EMPLOYEE]-(d:Department)
WHERE d.name = "IT Department"
RETURN p.name
```

Obiektowe bazy danych

- ▶ Wśród baz danych NoSQL jako pierwszy na rynku pojawił się mechanizm obiektowych baz danych, jednak z uwagi na znaczną popularność ustandaryzowanego języka SQL oraz wąskich specjalizacji pracowników sektora IT, nie udało się go spopularyzować.
- ▶ Główną przyczyną był podział obowiązków między osoby zajmujące się stricte bazami danych a programistów.
- ▶ Z punktu widzenia języków programowania, żonglowanie strukturami opartymi o obiekty wydaje się być najbardziej naturalne i logiczne
- ▶ Obiektowe systemy zarządzania bazą danych zapewniają tradycyjną funkcjonalność baz danych, lecz bazują na modelu obiektowym. Ich atutem jest udostępnianie danych w postaci obiektowej, czyli takiej samej w jakiej dane są przechowywane w programach napisanych w obiektowych językach programowania. Znika konieczność mapowania między modelem obiektowym a modelem relacyjnym jak to ma miejsce w przypadku użycia relacyjnej bazy danych.
- ▶ Przykładem bazy danych obiektowych jest **db4o**.
- ▶ Standard baz zorientowanych obiektowo jest dopiero tworzony, głównie przez Object Data Management Group (ODMG).

Własności obiektowego modelu danych

- ▶ Zachowanie obiektu opisuje się zbiorem metod – procedur operujących na stanie obiektu.
- ▶ Każdy obiekt jest jednoznacznie identyfikowany systemowym identyfikatorem (OID).
- ▶ Proste definiowanie obiektów złożonych.
- ▶ Obiekty o takich samych własnościach i zachowaniu grupuje się w klasy. Obiekt może być egzemplarzem tylko jednej klasy lub wielu.
- ▶ Klasy łączy się w hierarchie dziedziczenia, w których podklasa *dziedziczy* własności i metody nadklasy, dokładając swoje specyficzne.
- ▶ Niektóre modele pozwalają na przestąnianie (overriding) – zmianę odziedziczonej własności lub metody.

Różnice w stosunku do modelu relacyjnego

- ▶ Atrybuty mogą być wielowartościowe.
 - ▶ Eliminuje to potrzebę używania większości złączeń równościowych.
- ▶ Możliwość definiowania związków odwrotnych.
- ▶ Nie trzeba definiować kluczy, ich rolę pełnią OIDy.
 - ▶ Eliminuje to modyfikowanie wartości klucza.
- ▶ Dwa rodzaje równości: identyczność i równość wartości.
- ▶ Złączenia używane gdy warunki w zapytaniu dotyczą porównywania wartości atrybutów. W przypadku „kluczy zewnętrznych” bezpośrednia nawigacja z użyciem OID.
- ▶ Przy ładowaniu powiązanych obiektów z bazy danych do pamięci (buforowanie) OIDy zastępuje się wskaźnikami (pointer swizzling), co wielokrotnie przyspiesza nawigację.
- ▶ Wersjonowanie (długie transakcje) – tylko w niektórych OBD.

Wersje obiektów

- ▶ Potrzebne np. w systemach CAD do eksploracji wariantów.
- ▶ Każda wersja ma własny OID, ale zachowujemy *związki wyprowadzenia* między wersjami, najczęściej tworzące hierarchię.
- ▶ Hierarchia wersji zwykle trzymana w specjalnym *obiekcie generycznym*.
- ▶ Dwa rodzaje odwołań do wersji
 - ▶ *specyficzna* (albo *statyczna*): konkretna wersja
 - ▶ *generyczna* (albo *dynamiczna*): do *domyślnej* wersji (zwykle ostatniej).

Problemy z OBD (1)

- ▶ Brak optymalizacji zapytań. Główny problem to wyrażenia ścieżkowe, trudno dla nich budować indeksy.
- ▶ Brak dobrej formalizacji, ale są już algebry podobne do algebry relacji.
- ▶ Pięć typowych operacji: suma (union, różnica (difference), selekcja (select), generowanie (generate), odwzorowanie (map).
- ▶ Jednak brak powiązania tych operacji z niskopoziomowymi operacjami fizycznymi (takiego jak w algebrze relacji RBD), stąd ich arbitralność.
- ▶ Brak uniwersalnych języków zapytań. Zwykle brak zagnieżdżonych podzapytań, funkcji agregujących, grupowania. Brak automatycznego wsparcia dla obsługi ekstensji klasy – zbioru jej egzemplarzy; użytkownik musi sam zdefiniować *kolekcję* (collection) i pilnować jej aktualizacji przy dodawaniu i usuwaniu obiektów.
- ▶ Kompozycyjność
 - ▶ W modelu relacyjnym wynik zapytania jest (anonimową) relacją.
 - ▶ Można go więc obrobić kolejnym zapytaniem, co umożliwia składanie zapytań.
 - ▶ W modelu obiektowym jest gorzej, obiekty ze zbioru wynikowego mogą nie należeć do żadnej klasy.

Problemy z OBD (2)

- ▶ Brak wsparcia dla redefiniowania klas (odpowiednik ALTER z SQL), np. dodawania lub usuwania atrybutów.
- ▶ Brak sensownego mechanizmu definiowania uprawnień, nie wiadomo jaka ziarnistość: obiekt, zbiór, klasa, fragment hierarchii?
- ▶ Bardzo ograniczone więzy spójności, konieczność realizacji metodami.
- ▶ Kłopoty z współbieżnością, ręczne operowanie blokadami.
 - ▶ Problem długich transakcji (unika się ich w systemach OLTP dla relacyjnych baz danych).
 - ▶ Propozycja: wspólna publiczna baza danych + prywatne bazy danych użytkowników.

Pytania?