EKONOMICKÁ UNIVERZITA
Fakulta hospodárskej informatiky
Katedra aplikovanej informatiky

NoSQL- databázy

Meno : Bc. Martin Jankech
Stupeň/ročník: druhý/2.ročník
Školský rok: 2022/2023
Meno prednášajúceho: Ing. Jaroslav Kultán PhD.
Meno cvičiaceho: SCHMIDT, Peter, Ing. Mgr., PhD.

Obsah

Úvod

Internet sa za posledných 25 rokov rapídne zmenil, od prvého zoznamu online služieb, ktoré boli schopné ponúkať iba statické stránky obsahujúce novinky a správy až na dnešný variant, ktorý zahŕňa mobilné zariadenia, big data a rôzne typy iných aplikácií. Podľa prieskumu firmy IBM, každý deň sú ľudia schopní vyprodukovať 2.5 quintiliona bytov dát, viac ako 2 miliardy používateľov je aktívnych na sociálnych sieťach. Internet prezentuje nevídanú škálu možností a komplikuje tak dizajnérom život ohľadom inovatívnosti tvorby daných internetových služieb. Často si kladú otázku: Ako vytvoriť webový dizajn služby, ktorý bude responzívny, robustný, vždy dostupný, s nízkou latenciou (s nízkou čakacou dobou), vysokým výkonom schopným zniesť obrovský nápor tisícok používateľských požiadaviek. Kvôli narastajúcemu počtu používateľov a nárokov vecného obsahu internetových stránok je nutné, aby moderné webové služby dokázali ponúknuť dostupnosť pre čoraz väčšie množstvo používateľov, ako aj zabezpečiť obsluhu bez výpadku. Stojac tvárou v tvár týmto novým prekážkam, tradičná databázová technológia RDBMS nie je schopná držať krok s novými požiadavkami, a vďaka tomu na trh nastúpila nová alternatíva k už dovtedy používaným relačným databázam, a to konkrétne NoSQL (nerelačné databázy). Jedná sa o rozsiahlu triedu SRBD (Systémov riadenia bázy dát), ktoré nezodpovedajú relačným databázam pracujúcim na takej úrovni, že nie je nutné používať SQL príkazy na narábanie s údajmi uloženými v databáze. Táto databázová trieda chce spraviť prielom v rigidite relačného modelu, použitím viacerých modelov, ktoré dokážu uchovať údaje bez nutnosti definovať databázovú schému (Luo, Huang, 2013). Relačné databázy boli základom práce s aplikáciami niekoľko dekád, už od zverejnenia MySQL databázového systému v roku 1995 bolo MySQL veľmi populárnou a pomerne lacnou voľbou. A predsa len sa ale s explóziou vo výbere a rozmanitosti údajov za posledné roky podarilo novovzniknutým nerelačným databázovým technológiám preraziť na trhu ako alternatívna voľba na adresovanie k požiadavkám najnovších aplikácií (What is MongoDB?, 2017). NoSQL databázy, konkrétne Not Only SQL, databázy predstavujú alternatívne riešenie narábania s množinou dát k relačným databázovým systémom. Pod pojmom NoSQL si nemôžeme predstaviť len jeden konkrétny spôsob realizácie, ale skôr celok, ktorý zastrešuje veľké množstvo rôznorodých systémov uchovávania dát. NoSQL systémy získali popularitu kvôli rôznym faktorom, ku ktorým patrí napríklad flexibilita, ktorú dané systémy ponúkajú vývojárom pri tvorbe aplikácií. NoSQL sa totižto snaží o odpútanie sa od neohybnosti systému, ktoré práve ponúkajú relačné databázové systémy a ostatné štruktúrované dátové modely v prípade neštruktúrovaných a semištruktúrovaných dát.

1.0 Charakteristika pojmu NoSQL

V prvom rade je potrebné poznať význam a podstatu NoSQL. Anglický výraz NoSQL vznikol spojením dvoch slov Nie a SQL, čo napovedá tomu, že cieľom bolo úplne zamietnuť SQL. Aktuálne sa však väčšina prikláňa k tomu, že No je skratka NotOnly (nie iba) SQL. NoSQL nie je jeden produkt alebo jedna technológia, je to termín zastrešujúci celú triedu produktov, ktoré sa neriadia princípmi RDBMS. (Tiwari, 2011). NoSQL databázy vznikli ako interné riešenia reálnych problémov veľkých spoločností ako Amazon Dynamo, Google BigTable, LinkedIn Voldemort, Twitter FlockDB, Facebook Cassandra atď. Tieto spoločnosti začali riešiť 3 hlavné problémy:

- Vysoký objem transakcií,
- Znížiť latenciu prístupu k masívnym data-setom,
- Zvýšiť dostupnosť služieb v nespoľahlivom prostredí.

Spoločnosti sa najprv snažili tieto problémy vyriešiť pridaním výkonnejšieho hardvéru, neskôr skúšali zmenšiť relačné riešenia zjednodušovaním databázovej schémy, denormalizáciou schémy, znížením trvalosti a referenčnej integrity, cachovaním rôznych dopytov atď. Každá z techník čiastočne pomohla, ale nevyriešila, daný problém a zistili, že musia vybudovať riešenie na mieru. Veľa z týchto riešení bolo inšpiráciou pre mnohé aktuálne NoSQL riešenia na trhu (Burd, 2011).

2.0 História NoSQL databáz

NoSQL (Not Only SQL) databázy sa objavili ako alternatíva k relačným databázovým systémom na konci 20. storočia. História NoSQL databáz siaha až do 60. rokov, kedy sa objavili prvé hierarchické a sieťové databázy. Tieto systémy boli však veľmi komplexné a ťažko sa používali, čo viedlo k vývoju relačných databáz, ktoré sa stali dominantným typom databázových systémov.

S nástupom internetu a webových aplikácií sa ale ukázalo, že relačné databázy nie sú najlepšou voľbou pre ukladanie a spracovanie obrovského množstva nestruktúrovaných dát, ako sú napríklad sociálne siete, e-commerce weby alebo Big Data aplikácie. Z tohto dôvodu sa začali objavovať nové typy databázových systémov, ktoré boli navrhnuté tak, aby sa lepšie hodili pre prácu s neštruktúrovanými dátami a veľkými objemami dát.

Prvá NoSQL databáza bola dokumentová databáza Lotus Notes, ktorá bola uvedená na trh v roku 1989. V roku 2000 vznikla open-source databáza Apache Cassandra, ktorá bola

navrhnutá pre ukladanie veľkého množstva dát v distribuovanom prostredí. V roku 2004 potom vznikla kľúčovo-hodnotová databáza Redis, ktorá sa stala populárna pre cachovanie dát v reálnom čase.

Ďalšie významné NoSQL databázy vznikli v roku 2005, kedy sa objavila databáza CouchDB, ktorá bola navrhnutá pre ukladanie dokumentov vo formáte JSON, a v roku 2007, kedy vznikla dokumentová databáza MongoDB, ktorá sa stala veľmi populárnou pre ukladanie veľkého množstva neštruktúrovaných dát.

V posledných rokoch sa NoSQL databázy stali veľmi populárnymi pre ukladanie a spracovanie veľkého množstva dát v reálnom čase. V súčasnej dobe existuje veľa rôznych typov NoSQL databázových systémov, vrátane dokumentových databáz, kľúčovo-hodnotových databáz, grafických databáz a stĺpcovo orientovaných databáz.

(Litho, Mattsson, 2010).

3.0 Porovnanie SQL a NOSQL

Porovnanie SQL s NoSQL		
Oblasť	SQL	NoSQL
Podtypy	iba jeden typ	má viacero kategórií
Vynájdenie	vynájdené približne v roku 1970	moderné databázy 21. storočia
Škálovateľnosť	vertikálna	horizontálna
Konzistentný model	ACID transakčný model	skôr BASE model
Manipulácia	pomocou jazyka SQL	pomocou navrhnutých API
Variabilnosť dát	iba štrukturované dáta	všetky typy

(Litho, Mattsson, 2010).

4.0 Základné koncepty

Tradičné databázové technológie môžeme chápať ako spracovanie dát transakciami, sú charakterizované tzv. **ACID** vlastnosťami, t.j. atomicita (A), konzistencia (C), izolácia (I) a durabilita (D). V praxi, relačné databázy vždy plne podporovali ACID vlastnosti. Ak sa vzdialime od jednotlivých zložiek ACID, sme schopní docieliť omnoho vyšší výkon a rozšíriteľnosť. Existuje tzv. Eric Brewer's CAP teorém, ktorý zahŕňa konzistenciu (C), dostupnosť (A) a toleranciu segmentovania (P). Konzistencia znamená schopnosť všetkých uzlov vidieť tie isté dáta, v tom istom čase, dostupnosť predstavuje každú operáciu určujúcu ukončenie v určenej odozve. Nakoniec toleranciu segmentovania môžeme popísať ako schopnosť systému pokračovať v operácii napriek tomu, že by sieť prestala posielať správy medzi dvoma setmi serverov. Podľa CAP teorému, distribuovaný systém nedokáže uspokojiť všetky tri vlastnosti simultánne, ale iba dve. Nerelačné databázy vo všeobecnosti odstránili konzistenciu ako takú a využívajú tzv. **BASE**, čo je v preklade Basic availability (základná dostupnosť), Softstate (mäkký stav), Eventual consistency (výsledná konzistencia), ako náhradu za ACID. Zjednodušene by sa dalo povedať, že daná aplikácia funguje v podstate neustále (BA), nepotrebuje byť neustále konzistentná(S), ale bude sa nachádzať v istom stave (E) (Huang, 2013).

Tabuľka 1 Porovnanie vlastnosti jednotlivých modelov ACID vs. BASE1

ACID	BASE
Silná konzistencia	Slabá konzistencia
Izolovanosť	Dostupnosť na prvom mieste
Zameranie na „commit“	Zameranie na „Best effort“
Vnorené transakcie	Približné odpovede
Dostupnosť	Agresívnosť
Konzervatívna	Jednoduchšia
Náročné zmeny	Jednoduchší vývoj

(Huang, 2013).

5.0 Zdroje NoSQL databáz

NoSQL databázy sú navrhnuté tak, aby dokázali pracovať s rôznymi typmi dát, ktoré môžu byť štruktúrované, pološtruktúrované alebo úplne neštruktúrované. Preto existuje veľa rôznych zdrojov údajov, ktoré môžu byť použité pri práci s NoSQL databázami. Niektoré z týchto zdrojov zahŕňajú:

Webové stránky a sociálne médiá: NoSQL databázy sú často používané na ukladanie dát z webových stránok a sociálnych médií. Tieto dáta môžu byť v rôznych formátoch, ako sú napríklad JSON, XML alebo BSON.

Senzory a IoT zariadenia: NoSQL databázy sú často používané na ukladanie údajov z senzorov a IoT zariadení, ktoré môžu generovať veľké množstvá dát v reálnom čase. Tieto dáta môžu byť v rôznych formátoch, ako sú napríklad CSV alebo JSON.

Big Data a analytické dáta: NoSQL databázy sú často používané na ukladanie veľkých objemov dát, ktoré sa používajú na analytické účely. Tieto dáta môžu byť v rôznych formátoch, ako sú napríklad Apache Parquet, Apache Avro alebo JSON.

Mobilné aplikácie: NoSQL databázy sú často používané v mobilných aplikáciách na ukladanie údajov o používateľoch, ako sú napríklad profilové údaje alebo histórie nákupov. Tieto dáta môžu byť v rôznych formátoch, ako sú napríklad JSON alebo BSON.

E-mailové systémy: NoSQL databázy sa môžu použiť aj na ukladanie údajov z e-mailových systémov, ako sú napríklad inboxy, odozvy a prílohy. Tieto dáta môžu byť v rôznych formátoch, ako sú napríklad MIME alebo JSON.

Dáta z rôznych zdrojov: Ďalším zdrojom údajov pre NoSQL databázy môžu byť rôzne zdroje dát, ako sú napríklad súbory, relačné databázy alebo dokumenty v rôznych formátoch. NoSQL databázy často podporujú rôzne integrácie a importovacie nástroje, ktoré umožňujú importovať dáta z rôznych zdrojov do NoSQL databázy.

Celkovo existuje mnoho rôznych zdrojov údajov, ktoré môžu byť použité pri práci s NoSQL databázami. Dôležité je zvoliť vhodný formát údajov a zabezpečiť ich efektívne spracovanie a ukladanie v databáze. Rôzne NoSQL databázy sa líšia v tom, aké formáty dát podporujú a aké špeciálne funkcie a nástroje ponúkajú pre prácu s rôznymi typmi dát. (Huang, 2013).

4.0 Často používané dátové formáty pri NoSql databázach

Znalosť dátových formátov pri NoSQL databázach je neoddeliteľnou súčasťou v porovnaní s relačnými databázami, ktoré sú postavené na abstraktnom relačnom dátovom modeli, pretože pri NoSQL databázach formálny dátový model neexistuje alebo je jednoduchý a vnútornú štruktúru ukladaných dát rieši samotná aplikácia. Taktiež je dôležitá aj pri návrhu aplikácie postavenej na NoSQL databáze. Množstvo aplikácií v dnešnej dobe je postavených na JavaScripte a bežia na strane klienta vo webovom prehliadači. Výmena dát častí takýchto aplikácií sa realizuje v presne určenom formáte dát. Ideálny prípad je, keď je komunikačný

dátový formát rovnaký ako dátový formát pre ukladanie, aby sa obmedzila réžia potrebná pre konverziu dát (Holubová a kol., 2015).

Najznámejšími dátovými formátmi sú:

- **JSON** – skratka JSON (Javascript Object Notation) je formát dát odvodený z jazyka Javascript. JSON je často využívaný na výmenu dát medzi webovým prehliadačom a serverom. Tento dátový formát je textovou reprezentáciou štruktúrovaných dát, ukladaných do súborov párov kľúč hodnota. Takéto súbory sa volajú objekty, ktoré môžu byť hierarchické. JSON obsahuje malý súbor pravidiel, ktoré je potrebné dodržiavať. (Smith, 2015).
- **BSON** – je binárny spôsob zápisu JSON dokumentov. Výhody tohoto formátu oproti formátu JSON sú najmä v rýchlosti spracovania a taktiež disponuje špeciálnym dátovým typom pre dátum a čas a možnosťou ukladať binárne dáta (Holubová a kol., 2015).
- **XML** – je skratkou z anglického eXtensible Markup Language. XML je dátový formát, ktorý používa na štruktúrovanie dát značky. Dátový formát XML je jedným z najrozšírenejších dátových formátov, nakoľko je veľmi intuitívny a poskytuje možnosť predpisu formátu dát pomocou XML schém (Kosek, 2000).
- **YAML** – yaml (YAML Ain't Markup Language) nie je veľmi rozšíreným dátovým formátom a bol navrhnutý pre zrozumiteľný a človekom ľahko čitateľný zápis dát, pričom si stále zachováva možnosť mapovať dátové štruktúry používané v programovacích jazykoch.
- **CSV** – je skratkou pre Comma Separated Value. Jedná sa o triviálny dátový

formát, ktorý je podobný tabuľke. Obsahuje riadky, v ktorých sa nachádzajú jednotlivé položky oddelené čiarkami.

- **Linked Data** – je formát dát, ktorý sa používa prepojenie rôznych zdrojov dát na neskoršie dopytovanie týchto dát. Nadväzuje na myšlienky sémantického webu a dáta sú reprezentované pomocou dátového modelu RDF (Holubová a kol., 2015).

5.0 Základné typy NoSQL databáz

Dnešná doba ponúka veľké množstvo zástupcov NoSQL, a dokonca aj korporácie ako Google, či Amazon, pridali ruku k dielu a vytvorili hneď niekoľko nerelačných databáz pre ich vlastné potreby. Doposiaľ mohli byť dané databázy kategorizované podľa spôsobu, typu, formátu ukladania dát na:

- **Stĺpcové databázy**: záznamy sú uložené tak, aby boli rozšíriteľné a je možné ich rozdeliť vertikálne alebo horizontálne cez uzly. Takými sú napríklad Google Bigtable, Amazon SimpleDB, Hadoop HBase apod.
- **Databázy typu kľúč-hodnota**: v tzv. key-value systémoch, sú záznamy uchované ako hodnoty, ktoré môžu byť buď štruktúrované alebo naopak, úplne neštruktúrované, a zároveň unikátny kľúč identifikuje hodnotu, príkladmi sú Amazon Dynamo, Dynomite, Voldemort, Cassandra apod.
- **Dokumentovo-orientované databázy**: záznamy v danom systéme vyzerajú ako semi-štruktúrované dokumenty, ktoré sú vylepšené ešte o indexy. Tieto systém taktiež využívajú dotazovací mechanizmus na hľadanie záznamov, ako príklady môžeme uviesť CouchDB, MongoDB, OrientDB, SimpleDB apod.
- **Grafové databázy**: tieto systémy využívajú na svoju činnosť grafové modely s uzlami, hranami a nastaveniami na reprezentáciu a úschovu údajov, sú index free (bez indexov), každý element má priamy ukazovateľ na priľahlý element.Neo4j, OrientDB a InfiniteGraph sú typickými zástupcami danej skupiny systémov. (Holubová a kol., 2015).

5.1 Databázy kľúč – hodnota

Databázy typu kľúč hodnota umožňujú ukladať hodnoty definované kľúčom. Tieto databázy sú populárne práve pre rýchlu a jednoduchú manipuláciu s dátami. Model tejto databázy je podobný asociatívnemu poľu, v ktorom máme jedinečné kľúče, ktoré nám definujú uložené hodnoty. Tieto databázové systémy v základe poskytujú iba základné tri základné operácie API a to:

- PUT – pridáva nové páry kľúč–hodnota do databázy a aktualizuje záznamy

v prípade, že kľúč je už v databáze prítomný,

- GET – funkcia vráti hodnotu dopytovaného kľúča, prípadne môže vrátiť chybovú

hlášku pokiaľ sa kľúč v databáze nenachádza,

- DELETE – pomocou tejto funkcie môžeme zmazať kľúč s priradenou hodnotou

z tabuľky a taktiež v prípade neprítomnosti požadovaného kľúča v tabuľke môže funkcia vrátiť chybovú hlášku. Nevýhodou týchto databázových systémov je neschopnosť vyhľadať položky pomocou hodnoty ale iba pomocou kľúča, no v prípade podpory sekundárnych indexov vieme uložené dáta dopytovať aj pomocou týchto indexov. Naopak výhodou týchto

databázových systémov je rýchlosť a distribúcia dát medzi viacerými uzlami. Kľúče v týchto databázových systémoch sú jedinečným identifikátorom k priradenej hodnote tohoto kľúča. Obmedzenia kľúča závisia od konkrétneho DBMS, ale z dôvodu výkonu by nemal byť kľúč veľmi dlhý. (Holubová a kol., 2015).

Hodnota kľúča môže obsahovať údaje rôznych typov napr. krátky aj dlhý text, zdrojový kód, číslo, obrázok atď. Hodnotou môže byť aj vnorený objekt páru kľúč hodnota. Niektoré databázové systémy umožňujú určiť dátové typy údajov ako napr. Redis, ktorý disponuje možnosťou nastaviť dátový typ napr. reťazec, zoznamy, sady, triedené sady. atď. Na základe určenia dátových typov hodnôt môže databázový systém poskytnúť pokročilé funkcie ako napr. aktualizáciu viacerých polí, súčet alebo rozdiel pri práci s kolekciami.

Databázy typu kľúč hodnota sú často využívané ako cache systémy, ktoré si ukladajú často používané údaje do pamäte, aby sa eliminovalo množstvo prístupov na disk, čím dosiahneme vyššiu rýchlosť pri práci s údajmi. V tejto kategórii je niekoľko riešení od jednoduchých mapových štruktúr až po robustné systémy s politikou vypršania platnosti cache. Cache systémy sú často využívané na všetkých úrovniach počítačových

softvérov pre zvýšenie výkonu. Robustné distribuované systémy ako napr. EHCache sú často používané v Java aplikáciách. Ďalším populárnym cache systémom je Memcached, ktorý je často využívaný v oblasti webových aplikácií. Databázové systémy typu kľúč hodnota môžu byť užitočné pri ukladaní napr. stavoch správ, komentárov k článkom, obsahu košíka, vlastností produktov alebo dokonca aj na ukladanie celých webových stránok, pričom ako kľúč môžeme použiť URL webstránky. Medzi najznámejšie riešenia

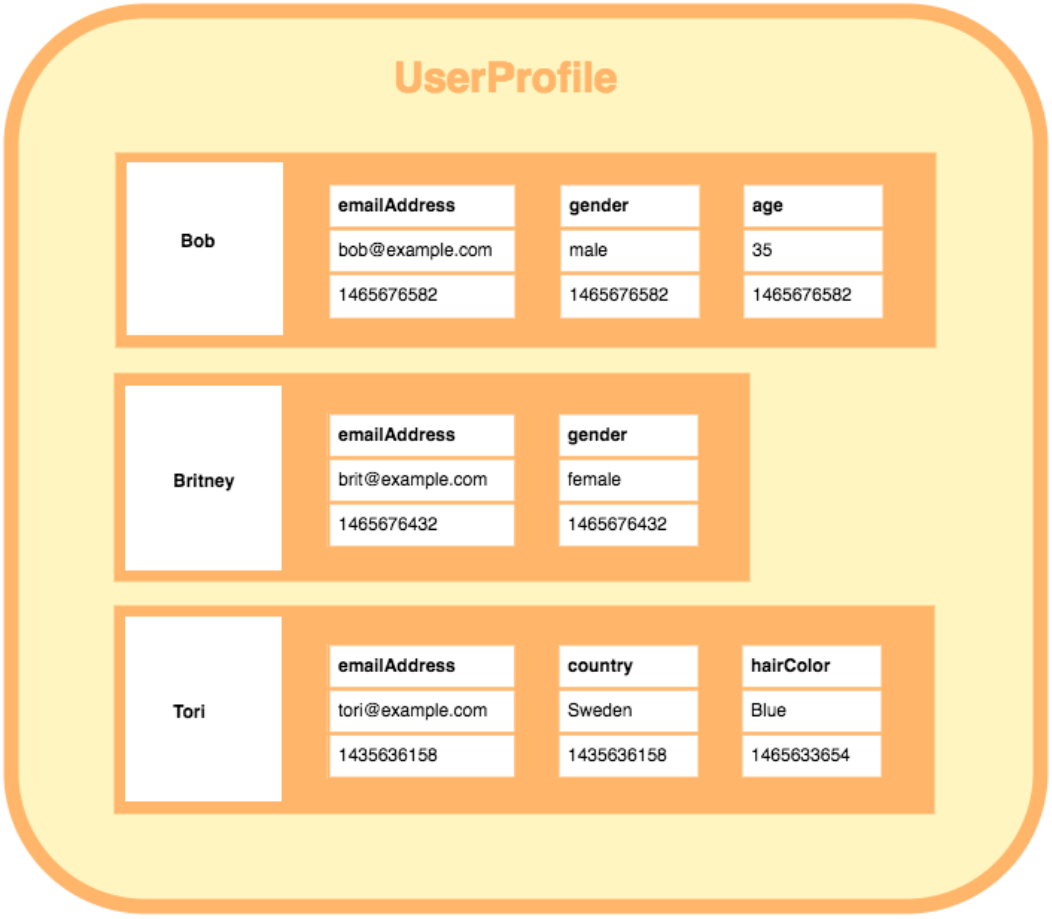
systémov typu kľúč hodnota paria:

- Redis,
- Memcached,
- Voldemort,
- Aerospike,
- Oracle Berkeley DB (Database.guide, 2016; Tiwari, 2011; Vaish, 2013).

5.2 Databázy rodiny stĺpcov

Databázy rodiny stĺpcov sú jedným z typov NoSQL databáz a v angličtine sú označované ako column family stores. Je potrebné odlíšiť ich od stĺpcovo orientovaných RDBMS (column oriented), ktoré sú označením pre relačné databázy, ukladajúce záznamy do stĺpcov relačných tabuliek ako napr. systémy MonetDB alebo Vertica (Holubová a kol.,2015). Väčšina databázových riešení typu rodiny stĺpcov bolo silne ovplyvnených projektom Goole

Bigtable ako napr. HBase, Hypertable alebo Cassandra, ktoré sa riadia modelom projektu Bigtable. Databázy rodiny stĺpcov ukladajú dáta vo forme stĺpcových rodín ako riadky, ktoré môžu obsahovať veľké množstvá stĺpcov. Rodiny stĺpcov sú vlastne skupiny súvisiacich údajov, ktoré bývajú často dopytované spolu. Databázy rodiny stĺpcov používajú identifikátory riadkov a stĺpcov ako univerzálne kľúče pre vyhľadávanie údajov. Každú položku dát vieme teda nájsť iba vtedy, pokiaľ poznáme identifikátory riadka a stĺpca. V porovnaní s RDBMS nemusíme ukladať všetky stĺpce pre všetky riadky a každý riadok tak môže mať rôzne stĺpce. Pri návrhu schém takýchto systémov, nám stačí navrhnúť iba rodiny stĺpcov. Najlepším spôsobom ako vizualizovať takto ukladané dáta je tabuľka (Database.guide, 2016; Tiwari, 2011; Vaish, 2013).



Obrázok 2 príklad rodiny stĺpcov s názvom UserProfile [2] Zdroj Obrázok 2: http://database.guide/what-is-a-column-store-database/

Základnými stavebnými prvkami databáz rodiny stĺpcov sú:

riadok – základný prvok modelu rodiny stĺpcov, ktorý je identifikovaný kľúčom riadku a môže obsahovať veľké množstvo stĺpcov,

stĺpec – stavebná jednotka riadku, ktorá obsahuje názov, hodnotu a časovú značku, kedy bola hodnota uložená. Každý riadok môže obsahovať rôzny počet stĺpcov,
super stĺpec – je stĺpec, ktorého hodnotu tvoria jeden alebo viacero podradených stĺpcov,
rodina stĺpcov – logická skupina súvisiacich riadkov a stĺpcov, pričom riadky v rôznych rodinách stĺpcov môžu mať rovnaký kľúč. Výhody týchto systémov sú predovšetkým v rýchlosti výberu veľkého množstva dát, nad ktorými vďaka štruktúre, v ktorej sa ukladajú, majú dobré výsledky aj pri použití agregačných funkcií. Ďalšou výhodou systémov rodiny stĺpcov je dobrá škálovateľnosť a možnosť využitia masívneho paralelného spracovania[3]. Medzi najznámejšie systémy rodí stĺpcov patrí Google Bigtable, Cassandra, Apache HBase, Hypertable alebo Druid

(Database.guide, 2016; Tiwari, 2011; Vaish, 2013).

5.3 Grafové databázy

Grafové databázové systémy sú efektívne na ukladanie objektov, vzťahov medzi objektmi a ich dopytovanie. Sú to systémy, ktoré obsahujú sekvenciu uzlov a vzťahov a vytvárajú tým graf (McCreary, Kelly,2014).

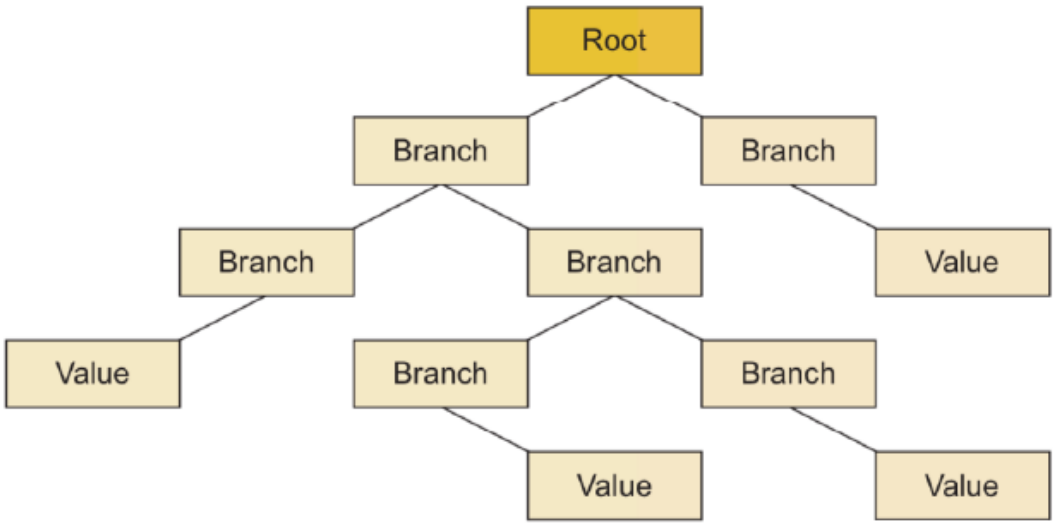
Uzlom v grafe sa myslí objekt, ktorý môže mať množinu atribútov. Vzťahy medzi objektmi nazývame hrany, ktoré môžu byť orientované a neorientované. Orientované hrany určujú vzťah jedného objektu k druhému, zatiaľ čo neorientované hrany vyjadrujú vzťah rovnako oboma smermi. Hrany rovnako ako uzly môžu mať rôzne atribúty a medzi dvomi objektami môžeme evidovať viacero vzťahov. V grafových databázach sa tiež môžeme stretnúť s pojmom cesta, ktorá nám vyjadruje počiatočný a koncový uzol a hrany medzi nimi. Dĺžku takejto cesty dostaneme súčtom hrán medzi začiatočným a koncovým uzlom. Grafy môžu byť jedno vzťahové, pri ktorých sú všetky hrany rovnakého typu alebo viac vzťahové, ktoré obsahujú hrany viacerých typov, na základe ktorého vieme rozlíšiť ich význam (Holubová a kol. 2015). Výhodou grafových databáz je rýchlosť, keďže tieto systémy si dokážu uchovávať grafové dáta v pamäti RAM, a tým sa znižuje počet prístupov na disk. Naopak, nevýhodou týchto databázových systémov je ich ťažké rozdelenie na viacero serverov pre úzke prepojenia každého uzla v grafe. Dáta je možné replikovať pre zlepšenie výkonnosti čítania, ale pri replikácii typu peer-to-peer môže byť implementácia zložitá. V grafovom modeli sa môžeme stretnúť so systémami Neo4j, Blazegraph, OrientDB, InfiniteGraph, Sparksee, Titan atď. Pri grafových databázach musíme spomenúť pojem RDF. Jedná sa o formát, ktorý bol publikovaný konzorciom W3C v roku 1999. RDF bol vyvinutý na spojenie dvoch

(Database.guide, 2016; Tiwari, 2011; Vaish, 2013).

externých množín dát vytvorených rôznymi organizáciami. Pomocou RDF môžeme načítať dve množiny údajov z externých zdrojov do jedného úložiska. Medzi najznámejšie RDF systémy patrí Virtuoso, 4store, Jena TDb atď (Holubová a kol., 2015).

5.4 Dokumentové databázy

- Dokumentové databázy, tiež označované ako dokumentovo orientované databázy, umožňujú vkladanie, vyhľadávanie a manipuláciu s čiastočne štruktúrovanými údajmi (Vaish, 2013). Semi-štruktúrované dáta sú dáta, ktoré majú určitú štruktúru, ale tá nemusí byť rigidná, pravidelná alebo úplná a všeobecne nemusia podliehať pevnej schéme (Connolly, Begg, 2005). Väčšina dokumentovo orientovaných databáz používa dátové formáty ako XML, JSON, BSON alebo YAML. Na prístup k údajom a manipuláciu s údajmi sa vo väčšine systémov používa RESTful API cez protokol HTTP alebo Apache Thrift protokol (Vaish, 2013). Je dôležité oddeliť tieto databázové systémy od systémov na správu dokumentov. Slovo dokumentové v názve dokumentové databázy reprezentuje spôsob ukladania dát po dokumentoch, teda skupín súvisiacich párov kľúč-hodnota. Dokumentové databázy spracovávajú dokument ako celok a zabraňujú jeho rozdeleniu na páry kľúčov a hodnôt, z ktorých sa skladá. Databázovo orientované systémy disponujú možnosťou uložiť viacero rôznorodých dokumentov do jednej kolekcie (Tiwari, 2011). Výhodou týchto databázových systémov je, že obsah záznamov nie je pevne definovaný žiadnou schémou, a tak môže každý záznam obsahovať iný súbor polí, čo je veľmi užitočné vo webových aplikáciách, kde je často potrebné uschovávať rôzne typy obsahu, ktoré sa môžu časom meniť. Aj napriek voľným schémam je možné vytvoriť a dopytovať indexy aj na základe vlastností dokumentu a nielen primárneho identifikátora (Vaish, 2013). Záznamy tak môžeme získať aj vyhľadaním ľubovoľnej hodnoty alebo obsahu v dokumente, pretože aj obsah dokumentu je indexovaný pri pridávaní nového dokumentu. Pretože dokumentovo orientované systémy používajú stromovú štruktúru, pri vyhľadávaní položiek vieme okrem existencie položky hľadaného výrazu, získať aj cestu k položke v stromovej štruktúre
  (Database.guide, 2016; Tiwari, 2011; Vaish, 2013).



Obrázok 3 Príklad ukladania podľa stromovej štruktúry v dokumentovo orientovaných databázach.

Obrázok 3 znázorňuje stromovú štruktúru ukladania v dokumentovo orientovaných databázach, ktorá začína koreňovým uzlom. Ten obsahuje vetvy stromu, ktoré môžu obsahovať ďalšie vetvy a poslednou sú listy, ktoré reprezentujú aktuálne hodnoty údajov(McCreary, Kelly, 2014). Medzi najpopulárnejšie riešenia dokumentovo orientovaných NoSQL databáz patrí MongoDB, CouchDB, MarkLogic, OrientDB (database.guide).

	Performance	**Scalability**	**Flexibility**	**Complexity**	**Functionality**
Key-value stores	High	High	High	None	Variable(none)
Column stores	High	High	Moderate	Low	Minimal
Document stores	High	Variable(high)	High	Low	Variable(low)
Graph datatabases	Variable	Variable	High	High	Graph theory
Related databases	Variable	Variable	Low	Moderate	Relational algebra

(Database.guide, 2016; Tiwari, 2011; Vaish, 2013).

6.0 Vybrané NOSQL systémy

6.1 HBase

HBase patrí medzi open source programy, je to nerelačný databázový model vytvorený ako časť Apache Software Foundation Hadoop projektu. Pracuje nad Hadoop Distributed File System (HDFS), a je vytvorený v jave. Poskytuje bezporuchový spôsob úschovy väčšieho objemu údajov ako je napríklad Bigtable za použitia Hadoop MapReduce funkcionality. HBase je vysoko výkonným, stĺpcovo-orientovaným, rozšíriteľne distribuovaným úložným systémom, a jeho tabuľky môžu byť použité aj ako vstup a aj ako výstup pre MapReduce procesy spustené v Hadoop. V HBase sa nachádzajú tri typy kľúčov: riadok, časová pečiatka a stĺpec. Riadok je primárnym kľúčom, dáta sú v tabuľke usporiadané podľa riadkového kľúča. Časová pečiatka reprezentuje čas každej vykonanej operácie, poukazuje na verziu dát v tabuľke. A napokon stĺpec je používaný ako úložisko atribútov patriacich k údajom, podporuje dynamický rozsah (Huang,2013).

6.2 Voldemort

Voldemort patrí medzi voľne dostupné key-value NoSQL systémy, používaný je napríklad známou webovou stránkou LinkedIn. Je to key-value databázový systém založený na princípe konzistencie hashovania. Dáta sú automaticky rozmiestnené na veľkom počte serverov. Údaje sú taktiež automaticky rozdelené tak, aby každý z daných serverov obsahoval iba podmnožinu z celkového množstva dát. Voldemort poskytuje zároveň laditeľnú konzistenciu. Každý modul vo Voldemorte zdieľa to isté prepojenie kódu, a každý modul má rozličnú funkciu. Vývojári sú schopní jednoducho zmeniť a zoskupiť tieto moduly podľa požiadaviek aplikácie(http://www.projectvoldemort. com/voldemort/).

6.3 Cassandra

Cassandra je špeciálny typ NoSQL, pretože ju môžeme nazvať do istej mierky hybridom medzi stĺpcovým kľúč-hodnota databázovým systémom, tento databázový systém používa napr. Facebook. Cassandra je založená na princípe Google Big Table a Amazon Dynamo. Najmenšia jednotka úschovy dát je stĺpec, s riadkami pozostávajúcimi zo stĺpcov a super stĺpcov. Cassandra podporuje SQL-like jazyk nazývaný CQL, spoločné s ostatnými protokolmi. Primárne a sekundárne indexy sú podporované, atomicita je garantovaná na úrovni jedného riadku tabuľky. Perzistencia je zaistená protokolovaním dát. Konzistencia je vysoko laditeľná podľa požadovaných operácií, čo v preklade znamená, že vývojár aplikácie môže stanoviť

požadovanú úroveň konzistencie, porovnaním latencie a konzistencie. Konflikty sú v danom databázovom systéme vyriešené na základe porovnania časových pečiatok (najnovší záznam je uchovaný) (Holubová a kol., 2015).

6.4 CouchDB

CouchDB je najznámejšia open source nerelačná databáza. Na ukladanie údajov a definovanie dokumentovo-orientovaného dátového modelu používa JSON. Neukladá dáta a vzťahy do tabuliek. Dáta sú zapuzdrované v dokumentovom formáte a samotná databáza v CouchDB je považovaná za kolekciu nezávislých dokumentov. Neexistuje žiadna normalizovaná schéma v danom databázovom systéme, dokument si udržuje svoje vlastné dáta a celistvú schému.

CouchDB pre svoju prácu využíva tzv. mechanizmus Multi-Version Concurrency Control (MVCC), aby sa vyhol uzamkýnaniu databázového súboru počas zapisovania nových údajov do danej databázy. Táto metóda zreteľne zlepšuje rýchlosť vstupu, pretože akékoľvek konflikty sú ponechané na konkrétnu aplikáciu, aby ich
vyriešila. Ďalšia hlavná charakteristika CouchDB je, že podporuje ACID vlastnosti, čím sa líši od ostatných dokumentovo-orientovaných databázových systémov (Huang, 2013).

6.5 SimpleDB

Amazon SimpleDB je vysoko dostupný NoSQL databázový systém, ktorý uľahčuje prácu s databázovou administratívou. Vývojári jednoducho ukladajú a vyhľadávajú údajové položky cez webové služby prostredníctvom požiadaviek a SimpleDB sa postará o zvyšok. SimpleDB je optimalizovaný tak aby poskytoval vysokú dostupnosť a flexibilitu s čo možno najmenšou administratívnou záťažou. Daný databázový systém vytvára a spravuje viacnásobne geograficky distribuované repliky uchovaných dát automaticky, aby zabezpečil vysokú dostupnosť a odolnosť. Dátový model môže byť zmenený aj počas priebehu a údaje budú automaticky indexované pre vývojára (Amazon SimpleDB, 2016).

6.6 MongoDB

MongoDB je dokumentovo-orientovaný databázový systém, ktorý je písaný v C++ a vyvinutý spoločnosťou 10gem. Používa JSON(dáta sú uložené a prenášané binárne vo vlastnom formáte BSON), umožňujúci vytvorenie tzv. Schemaless data modelu(Databázovému modelu bez schémy), kde jedinou podmienkou je iba existencia jedinečného identifikačného čísla ID.

Manipulácia s dokumentmi je hlavným zameraním MongoDB, keďže samotný databázový systém ponúka množstvo rôznych frameworkov a možností interakcie s danými dokumentmi. Dokumenty môžu byť dotazované, usporiadané, iterované s kurzorom, agregované apod. Zmeny v dokumentoch sú atomického typu (Lourenco et al., 2015).

6.7 CouchBase

Couchbase je kombinácia Membase (kľúč-hodnota systému s tzv. Memcached kompatibilitou) a CouchDB. Tento databázový systém môže byť používaný aj v štýle kľúč-hodnota, ale je často považovaný za predstaviteľa dokumentovo-orientovaných databázových systémov. Dokumenty v Couchbase majú vnútorný unikátny identifikačný kľúč a ten je uložený v tzv. data buckets. Rovnako ako v prípade CouchDB, aj tu sú dotazy vytvárané pomocou MapReduce v Javascripte. To čím sa hlavne líšia Couchbase a CouchDB, je vo fyzickom rozdelení tabuliek. CouchDB natívne nepodporuje takýto typ partície, avšak Couchbase prichádza s transparentným delením off-the-self s aplikačnou transparentnosťou. Ďalším rozdielom týchto databáz je replikácia. Couchbase podporuje tzv. interklastrové kopírovanie (Lourenco et al.,
2015).

6.8 Neo4j

Neo4j je open-source databáza, ktorá ukladá údaje do grafov. To v preklade znamená úschovu dát v dátovej štruktúre, ktorá je schopná jasne reprezentovať rozličné druhy dát vo vysokej prístupnosti. Údaje sú ukladané ako uzly a vzťahy. Aj údaje aj vzťahy sú schopné držať vlastnosti vo forme kľúč-hodnota. Hodnoty zobrazujúce vlastnosti môžu byť buď primitívne, alebo môžu byť poľom s jedným primitívnym typom dát. Uzly a vzťahy majú interne unikátne identifikátory, ktoré môžu byť použité ako možnosť vyhľadávania. Sémantika môže byť vyjadrená pridaním priamych vzťahov

medzi uzlami

(Lourenco et al., 2015).

7.0 Cloudové riešenia

Microsoft Azure a Amazon Web Services (AWS) sú dva najväčšie poskytovatelia cloudových služieb na svete. Oba poskytujú NoSQL databázy a príslušné služby. NoSQL databázy sú navrhnuté pre ukladanie a manipuláciu s neštrukturovanými alebo

pološtrokturovanými dátami, ktoré sa nedajú ľahko uložiť do relačnej databázy. NoSQL databázy sú veľmi flexibilné a umožňujú ukladanie rôznych typov dát, ako sú dokumenty, grafy, kľúč-hodnota páry alebo stĺpcové rodiny. Azure poskytuje NoSQL databázy, ako sú Azure Cosmos DB a Azure Table Storage. Azure Cosmos DB umožňuje ukladať a spracovávať dáta pomocou rôznych API, ako sú SQL, MongoDB, Cassandra a Gremlin. Azure Table Storage poskytuje jednoduchú tabuľkovú štruktúru pre ukladanie dát pomocou REST API. AWS poskytuje NoSQL databázy, ako sú Amazon DynamoDB a Amazon DocumentDB. Amazon DynamoDB je databáza typu kľúč-hodnota, ktorá umožňuje ukladanie a rýchle vyhľadávanie dát pomocou primárneho kľúča. Amazon DocumentDB je databáza dokumentov, ktorá umožňuje ukladanie a spracovanie dát v JSON formáte. Okrem NoSQL databáz poskytujú Azure a AWS aj rôzne služby, ako sú replikácia dát, zálohovanie a obnovenie, monitorovanie a škálovanie dátových skladov. Tieto služby umožňujú spravovať a optimalizovať NoSQL databázy pre rôzne použitia a zabezpečiť ich dostupnosť a výkon.

(Lourenco et al., 2015).

Záver

Z tejto seminárnej práce vyplýva, že NoSQL databázy sú dôležitou súčasťou moderného sveta databázových technológií. V porovnaní so štandardnými relačnými databázami poskytujú NoSQL databázy výhodu v rýchlosti a flexibilite. Tieto databázy umožňujú ukladať a spracovávať rôzne typy dát, vrátane nestruktúrovaných dát, a preto sú ideálne pre veľké množstvá dát a webové aplikácie, ktoré potrebujú horizontálne škálovanie.

V práci sme sa zaoberali charakteristikou pojmu NoSQL, históriou NoSQL databáz, porovnaním SQL a NoSQL, základnými konceptami, zdrojmi NoSQL databáz, dátovými formátmi a typmi NoSQL databáz. Popísali sme tiež niektoré populárne NoSQL databázové systémy a ich vlastnosti. Okrem toho sme sa zaoberali aj cloudovými riešeniami NoSQL databáz a ukázali sme, že cloudové NoSQL databázy poskytujú jednoduchší a efektívnejší spôsob, ako udržiavať a škálovať dáta.

Vzhľadom na rýchly vývoj technológií a neustále rastúce množstvo dát, NoSQL databázy budú mať v budúcnosti dôležitú úlohu pri ukladaní a spracovaní dát.

Použitá literatúra

HUANG YU, TEIJAN LUO 2013. NoSQL Database: A Scalable, Availability, High Performance Storage for Big Data. In. Pervasive Computing and the Networked World, Joint International Conference, ICPCA/SWS 2013, Vina del Mar, Chile, December 5-7, 2013. Revised Selected Papers,Zu Qiaohong,Vera Vargas Maria, Hu Bo (Eds.) [online]. 2016. [cit 2023-03-30] Dostupné na internete: <http://www.springer.com/us/book/9783319092645/>.

WHAT IS MONGODB? 2017. What is mongodb? [online]. 2017. [cit. 2023-03-30]. Dostupné na internete: <https://www.mongodb.com/what-is-mongodb>.

BURD, G. 2011. *NoSQL.* [online]. [cit. 2023-03-30]. Dostupné na internete: <http://static.usenix.org/publications/login/2011-10/openpdfs/Burd.pdf>.

LITH, A; MATTSON, J (2010). "Investigating storage solutions for large data: A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data" Göteborg: Department of Computer Science and Engineering, Chalmers University of Technology. p. 70. Retrieved 12 May 2011. Carlo Strozzi first used the term NoSQL in 1998 as a name for his open source relational database that did not offer a SQL interface[...]

HOLUBOVÁ, I, KOSEK, J., MINAŘÍK, K., NOVÁK, D. 2015. *Big Data a NoSQL databáze*. Grada Publishing, a.s. Praha. 2015. 288 s. ISBN 978-80-247-5938-8.

KOSEK, J. 2000. *XML pro každého*. Grada Publishing. Praha. 2000. 163 s. ISBN 80-7169-860-1.

SMITH, B. 2015. *Beginning JSON: Learn the preferred data format of the web*. Apress Media. California. 2015. 324 s. ISBN 978-1-4842-0202-9.

TIWARI, S. 2011. *Professional NoSQL*. John Wiley & Sons, Inc. Indianapolis, Indiana. 2011. 361 s. ISBN 978-0-470-94224-6.

LOURENCO, J et.al.(2015). Comparing NoSQL Databases with a Relational Database: Performance and Space. Services Transactions on Big Data. 2. 1-14. 10.29268/stbd.2015.2.1.1.