

# Podklady pro výuku multimodelové databáze Cosmos DB

Sergei Popov

---

Bakalářská práce  
2024

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: **Sergei Popov**  
Osobní číslo: **A21261**

Téma práce: **Podklady pro výuku multimodelové databáze Cosmos DB**  
Téma práce anglicky: **Lecture notes for Cosmos DB**  
Jazyk práce: **Čeština**

Vedoucí práce: **doc. Ing. Petr Šilhavý, Ph.D.**  
Ústav počítačových a komunikačních systémů

## Zásady pro vypracování:

1. Seznamte se s Cosmos DB v aktuální verzi.
2. Zpracujte podklady pro cvičení a přednášky v rozsahu min. 10 týdnů.
3. Realizujte sadu vzorových příkladů.
4. Navrhněte sadu testovacích úkolů včetně řešení ve formátu vhodném pro moodle.
5. Uveďte možné směry rozvoje učební pomůcky.

## Seznam doporučené literatury:

1. PAZ, José Rolando Guay. Introduction to azure cosmos db. In: *Microsoft Azure Cosmos DB Revealed*. Apress, Berkeley, CA, 2018. p. 1-23.
2. HILLAR, Gaston C.; YÖNDEM, Daron. *Guide to NoSQL with Azure Cosmos DB: Work with the massively scalable Azure database service with JSON, C#, LINQ, and .NET Core 2*. Packt Publishing Ltd, 2018.
3. Dokumentace k Azure Cosmos DB, 2020. Dokumentace k Azure Cosmos DB [online]. USA: Microsoft [cit. 2020-11-25]. Dostupné z: <https://docs.microsoft.com/cs-cz/azure/cosmos-db/>
4. OWLER, Adam. *NoSQL for dummies*. John Wiley & Sons, 2015.
5. FREEMAN, Adam. Putting ASP. NET Core in Context. In: *Pro ASP. NET Core 3*. Apress, Berkeley, CA, 2020. p. 3-7.
6. LOFF, Brian. *Azure Cosmos DB and DocumentDB. A Complete Guide – 2021 Edition*. Emereo Publishing, 2021.
7. VERBEEK, Marijn; RUNDQUIST, Isaac. *CosmosDB On The Go: Learn Azure Cosmos DB for Mobile Developers*. Apress, 2020.
8. JASSAL, Parikshit. *Azure Cosmos DB for MongoDB Developers: Migrating to a Globally Distributed Database*. Apress, 2018.
9. KLOSTER, Dan. *Cosmos DB for SQL Server Developers: Migrating to NoSQL from SQL*. Apress, 2017.
10. CHANDER, Priyank. *Azure Cosmos DB for Real-Time Analytics and IoT: The Complete Guide*. Apress, 2021.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum:

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Sergei Popov

## ABSTRAKT

V dnešní době, kdy digitální technologie pronikají do všech aspektů našeho života, se stává zpracování a analýza dat klíčovou kompetencí pro mnoho odvětví. Jedním z přístupů, který nabízí řešení pro efektivní manipulaci s rozsáhlými objemy dat, je využití cloudových databázových technologií. Microsoft Azure Cosmos DB, představující multimodelovou databázovou platformu, se v tomto kontextu jeví jako inovativní nástroj schopný adresovat specifické požadavky aplikací vyžadujících vysokou dostupnost a globální distribuci dat.

Cílem této bakalářské práce je nejenom seznámit se s aktuální verzí Cosmos DB, ale především vytvořit podklady, které by mohly sloužit pro výuku této moderní technologie. Práce zahrnuje tvorbu vzorových příkladů, cvičení a testovacích úkolů, které pomohou studentům lépe porozumět a prakticky využívat Cosmos DB ve svých budoucích projektech.

Multimodelová schopnost Cosmos DB umožňuje vývojářům používat různé datové modely podle potřeb jejich aplikací, což činí tuto platformu extrémně flexibilní. V průběhu této práce jsou probrány klíčové koncepty jako globální distribuce, konzistence dat, modely konzistence a strategie škálování, které jsou zásadní pro optimalizaci výkonu a efektivity databázových operací v cloudovém prostředí.

Výukový plán navržený v této práci zahrnuje teoretické lekce a praktická cvičení rozdělená do deseti týdnů, postupující od základních konceptů až po pokročilé funkce Cosmos DB. Studenti se tak naučí nejen teoretické znalosti, ale i praktické dovednosti prostřednictvím cvičení a projektových úkolů.

Nakonec, tato práce navrhuje možné směry rozvoje výukového plánu, což zahrnuje integraci s dalšími službami Azure, práci s reálnými datovými sadami a rozšíření o nové moduly zahrnující pokročilé techniky databázového managementu. Předpokládá se, že výsledky této práce pomohou studentům technických oborů získat hlubší porozumění cloudovým databázovým technologiím a jejich využití v praxi.

Tato bakalářská práce poskytuje komplexní přehled o Azure Cosmos DB a slouží jako základ pro výuku této technologie, což přispěje k rozvoji dovedností studentů v oblasti moderních databázových systémů.

Klíčová slova:

Azure Cosmos DB, cloudové databáze, multimodelová databáze, CRUD operace, škálování, konzistence dat, výukový plán, Microsoft Azure, globální distribuce dat, datové modely,

NoSQL, dokumentově orientovaný model, grafový model, model klíč-hodnota, sloupcově orientovaný model, bezpečnost dat, správa přístupu, monitorování výkonu, optimalizace dotazů, šifrování, compliance, vývojářské nástroje, serverless architektura, umělá inteligence, migrace dat, zálohování, automatizace škálování.

## ABSTRACT

Today, as digital technologies permeate all aspects of our lives, data processing and analysis are becoming key competencies for many industries. One approach that offers solutions for efficiently handling large volumes of data is using cloud database technologies. Microsoft Azure Cosmos DB, representing a multimodel database platform, appears in this context as an innovative tool capable of addressing the specific requirements of applications requiring high availability and global data distribution.

The aim of this bachelor thesis is not only to get acquainted with the current version of Cosmos DB, but also to create materials that could serve to teach this modern technology. The work includes creating sample examples, exercises and test tasks to help students better understand and use Cosmos DB in future projects.

The multi-model capability of Cosmos DB allows developers to use different data models according to the needs of their applications, which makes this platform extremely flexible. This work discusses key concepts such as global distribution, data consistency, consistency models and scaling strategies, which are essential for optimising the performance and efficiency of database operations in a cloud environment.

The curriculum proposed in this work includes theoretical lessons and practical exercises divided into ten weeks, progressing from basic concepts to advanced Cosmos DB functions. Thus, students learn theoretical knowledge and practical skills through exercises and project tasks.

Finally, this work suggests possible directions for developing the tutorial, which includes integration with other Azure services, work with real data sets and expansion with new modules incorporating advanced database management techniques. It is expected that the results of this work will help students of technical disciplines gain a deeper understanding of cloud database technologies and their use in practice.

This bachelor thesis provides a comprehensive overview of Azure Cosmos DB. It serves as a basis for teaching this technology, which will contribute to developing students' skills in modern database systems.

Keywords:

Azure Cosmos DB, cloud databases, multimodel database, CRUD operations, scaling, data consistency, tutorial, Microsoft Azure, global data distribution, data models, NoSQL,

document-oriented model, graph-based model, key-value model, column-based model, data security, access management, performance monitoring, query optimization, encryption, compliance, developer tools, serverless architecture, data migration, backup, automation scaling.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Prohlašuji, že jsem při vypracování této bakalářské práce využil nástroje umělé inteligence, konkrétně jazykový model ChatGPT od společnosti OpenAI, který mi poskytl podporu v oblastech vyhledávání informací, generování textu a úprav obsahu. Přestože byly některé části práce vytvořeny za použití těchto nástrojů, veškerý obsah byl pečlivě zkontrolován a upraven tak, aby splňoval akademické standardy a požadavky mého studijního programu.

Tímto potvrzuji, že veškeré informace, analýzy a závěry uvedené v této práci jsou mé vlastní a odpovídají stanoveným etickým normám a zásadám akademického psaní.



**OBSAH**

<b>OBSAH .....</b>	<b>9</b>
<b>ÚVOD.....</b>	<b>12</b>
<b>TEORETICKÁ ČÁST.....</b>	<b>13</b>
<b>SEZNÁMENÍ S COSMOS DB .....</b>	<b>14</b>
<b>1.1 HISTORIE VÝVOJE COSMOS DB.....</b>	<b>14</b>
<b>1.2 SLUŽBY AZURE OD MICROSOFTU .....</b>	<b>14</b>
<b>1.3 ARCHITEKTURA A KLÍČOVÉ VLASTNOSTI .....</b>	<b>15</b>
<b>1.4 MODELY DAT A JEJÍCH POUŽITÍ.....</b>	<b>16</b>
1.4.1 DOKUMENTOVĚ ORIENTOVANÝ MODEL (NoSQL API) .....	17
1.4.2 GRAFOVÝ MODEL (GREMLIN API) .....	17
1.4.3 MODEL KLÍČ-HODNOTA (TABLE API) .....	17
1.4.4 SLOUPCOVĚ ORIENTOVANÝ MODEL (CASSANDRA API) .....	17
<b>1.5 KLÍČOVÉ KONCEPTY .....</b>	<b>17</b>
<b>2 STRATEGIE EFEKTIVNÍHO PLÁNOVÁNÍ VÝUKY S DŮRAZEM NA COSMOS DB.....</b>	<b>19</b>
<b>2.1 ZÁKLADY PLÁNOVÁNÍ VÝUKY IT A COSMOS DB .....</b>	<b>19</b>
<b>2.2 STANOVENÍ VZDĚLÁVACÍCH CÍLŮ .....</b>	<b>20</b>
<b>2.3 VÝBĚR OBSAHU A STRUKTURACE MATERIÁLŮ.....</b>	<b>21</b>
2.3.1 MODULARITA .....	21
2.3.2 SEKVENCOVÁNÍ.....	21
2.3.3 INTEGRACE VIZUÁLNÍCH POMŮCEK .....	22
2.3.4 INTERAKTIVNÍ KOMPONENTY .....	22
2.3.5 ZPĚTNÁ VAZBA A HODNOCENÍ.....	22
2.3.6 DOPROVODNÉ MATERIÁLY .....	22
2.3.7 PŘÍSTUPNOST A ADAPTABILITA .....	22
<b>2.4 PEDAGOGICKÉ METODY A TECHNIKY VÝUKY IT .....</b>	<b>22</b>
2.4.1 PROJEKTOVĚ ORIENTOVANÉ UČENÍ (PROJECT-BASED LEARNING, PBL) .....	23
2.4.2 FLIPPED CLASSROOM .....	23
2.4.3 KOLABORATIVNÍ UČENÍ .....	23
2.4.4 ONLINE TUTORIÁLY A MOOCs .....	23
2.4.5 ADAPTIVNÍ UČENÍ.....	23
<b>2.5 HODNOCENÍ A ZPĚTNÁ VAZBA.....</b>	<b>24</b>
2.5.1 FORMY HODNOCENÍ.....	24
2.5.2 TECHNIKY ZPĚTNÉ VAZBY .....	24
<b>PRAKTICKÁ ČÁST .....</b>	<b>25</b>
<b>3 STRUKTUROVANÝ PLÁN PRO VÝUKU.....</b>	<b>26</b>
<b>3.1 PŘEDNÁŠKY .....</b>	<b>26</b>

3.1.1	FORMÁT A OBSAH PŘEDNÁŠEK .....	26
3.1.2	OBSAHOVÁ NÁPLŇ PŘEDNÁŠEK .....	27
3.1.3	CÍLE PŘEDNÁŠEK .....	27
3.1.4	ZÍSKANÉ ZNALOSTI.....	27
<b>3.2</b>	<b>PREZENTACÍ PRO PŘEDNÁŠKY .....</b>	<b>28</b>
3.2.1	1. TÝDEN: ÚVOD DO AZURE COSMOS DB.....	28
3.2.2	2. TÝDEN: DATOVÉ MODELY A API .....	29
3.2.3	3. TÝDEN: KONZISTENCE DAT .....	30
3.2.4	4. TÝDEN: ŠKÁLOVÁNÍ A DOSTUPNOST .....	31
3.2.5	5. TÝDEN: INDEXACE A DOTAZOVÁNÍ .....	32
3.2.6	6. TÝDEN: BEZPEČNOST A SPRÁVA PŘÍSTUPU.....	34
3.2.7	7. TÝDEN: MONITOROVÁNÍ A LADĚNÍ VÝKONU.....	35
3.2.8	8. TÝDEN: INTEGRACE S DALŠÍMI AZURE SLUŽBAMI.....	36
3.2.9	9. TÝDEN: MIGRACE DAT DO COSMOS DB.....	37
3.2.10	10. TÝDEN: CASE STUDIES A BEST PRACTICES.....	38
<b>3.3</b>	<b>CVIČENÍ.....</b>	<b>40</b>
3.3.1	POPIS PRŮBĚHU CVIČENÍ.....	40
3.3.2	OBSAH CVIČENÍ.....	40
3.3.3	ZÁPOČTOVÝ ÚKOL.....	41
<b>3.4</b>	<b>PREZENTACÍ PRO PRAKTICKÉ CVIČENÍ.....</b>	<b>41</b>
3.4.1	1. TÝDEN: ÚVOD DO PROSTŘEDÍ AZURE A VYTVOŘENÍ ÚČTU AZURE.....	41
3.4.2	2. TÝDEN: VYTVOŘENÍ ÚČTU COSMOS DB .....	42
3.4.3	3. TÝDEN: ÚVOD DO ZÁKLADNÍCH KONCEPTŮ A ARCHITEKTURY COSMOS DB .....	43
3.4.4	4. TÝDEN: ŠKÁLOVÁNÍ A DOSTUPNOST + ZADÁNÍ PROJEKTU .....	44
3.4.5	5. TÝDEN: INSTALACE A NASTAVENÍ NÁSTROJŮ PRO PRÁCI S AZURE COSMOS DB 45	
3.4.6	6. TÝDEN: PŘIDÁVÁNÍ DAT DO AZURE COSMOS DB (CREATE).....	46
3.4.7	7. TÝDEN: AKTUALIZACE DAT V AZURE COSMOS DB (UPDATE).....	47
3.4.8	8. TÝDEN: MAZÁNÍ DAT V AZURE COSMOS DB (DELETE).....	48
3.4.9	9. TÝDEN: ČTENÍ DAT A PRÁCE S DATY V AZURE COSMOS DB POMOCÍ JAVASCRIPTU (READ) .....	49
3.4.10	10. TÝDEN: OBHAJOBA PROJEKTŮ .....	50
<b>3.5</b>	<b>ABSOLVOVÁNÍ PŘEDMĚTU .....</b>	<b>51</b>
3.5.1	KLASIFIKOVANÝ ZÁPOČET .....	51
3.5.2	ZADÁNÍ PRO STUDENTY: VYTVOŘENÍ MALÉ DATABÁZE S CRUD OPERACEMI V AZURE COSMOS DB .....	53
<b>4</b>	<b>CASE STUDY: NEWDAY A IMPLEMENTACE AZURE COSMOS DB .....</b>	<b>55</b>
<b>4.1</b>	<b>ÚVOD .....</b>	<b>55</b>
<b>4.2</b>	<b>VÝZVY A CÍLE .....</b>	<b>55</b>
<b>4.3</b>	<b>ŘEŠENÍ: INTEGRACE AZURE COSMOS DB.....</b>	<b>55</b>
<b>4.4</b>	<b>IMPLEMENTACE A TECHNICKÉ ÚPRAVY .....</b>	<b>56</b>
<b>4.5</b>	<b>VÝSLEDKY A DOPADY.....</b>	<b>56</b>

4.6 ZÁVĚR.....	56
5 TESTOVACÍ ÚKOLY PRO AZURE COSMOS DB - MOODLE FORMÁT.....	57
5.1 ÚKOL 1: VLOŽENÍ DOKUMENTU.....	57
5.2 ÚKOL 2: AKTUALIZACE DOKUMENTU .....	57
5.3 ÚKOL 3: SMAZÁNÍ DOKUMENTU .....	58
5.4 ÚKOL 4: ČTENÍ DAT POMOCÍ SQL DOTAZŮ.....	59
5.5 ÚKOL 5: MIGRACE DAT DO AZURE COSMOS DB.....	59
6 TESTOVACÍ OTÁZKY Z TEORIE AZURE COSMOS DB .....	60
6.1 1. TÝDEN: ÚVOD DO AZURE COSMOS DB .....	60
6.2 2. TÝDEN: DATOVÉ MODELY A API.....	60
6.3 3. TÝDEN: KONZISTENCE DAT .....	61
6.4 4. TÝDEN: ŠKÁLOVÁNÍ A DOSTUPNOST .....	61
6.5 5. TÝDEN: INDEXACE A DOTAZOVÁNÍ.....	62
6.6 6. TÝDEN: BEZPEČNOST A SPRÁVA PŘÍSTUPU.....	62
6.7 7. TÝDEN: MONITOROVÁNÍ A LADĚNÍ VÝKONU .....	63
6.8 8. TÝDEN: INTEGRACE S DALŠÍMI AZURE SLUŽBAMI .....	63
6.9 9. TÝDEN: MIGRACE DAT DO COSMOS DB.....	64
6.10 10. TÝDEN: CASE STUDIES A BEST PRACTICES.....	64
7 VZOROVÝ PŘÍKLAD + TUTORIÁL .....	66
7.1 WEB PRO SPRÁVU DATABÁZE ZÁKAZNÍKŮ .....	66
7.2 TUTORIÁL.....	67
7.3 CO JE TO PUG, EXPRESS A NPM? .....	73
7.3.1 PUG74	
7.3.2 EXPRESS.....	74
7.3.3 NPM75	
8 MOŽNÉ SMĚRY ROZVOJE VÝUKOVÉHO PLÁNU .....	76
8.1 POKROČILÉ DOTAZOVÁNÍ A INDEXACE.....	76
8.2 PRÁCE S RŮZNÝMI DATOVÝMI MODELY.....	76
8.3 BEZPEČNOST A SPRÁVA PŘÍSTUPU .....	76
8.4 MONITOROVÁNÍ A LADĚNÍ VÝKONU .....	76
8.5 SPRÁVA A ŠKÁLOVÁNÍ.....	76
8.6 MIGRACE DAT A ZÁLOHOVÁNÍ.....	77
8.7 INTEGRACE S DALŠÍMI SLUŽBAMI AZURE.....	77
ZÁVĚR .....	78
SEZNAM POUŽITÉ LITERATURY.....	80
SEZNAM OBRÁZKŮ .....	82
SEZNAM PŘÍLOH.....	83

## ÚVOD

V dnešní době, kdy digitální technologie pronikají do všech aspektů našeho života, se stává zpracování a analýza dat klíčovou kompetencí pro mnoho odvětví. Jedním z přístupů, který nabízí řešení pro efektivní manipulaci s rozsáhlými objemy dat, je využití cloudových databázových technologií. Microsoft Azure Cosmos DB, představující multimodelovou databázovou platformu, se v tomto kontextu jeví jako inovativní nástroj schopný adresovat specifické požadavky aplikací vyžadujících vysokou dostupnost a globální distribuci dat.

Cílem této bakalářské práce je nejenom seznámit se s aktuální verzí Cosmos DB, ale především vytvořit podklady, které by mohly sloužit pro výuku této moderní technologie. Práce bude zahrnovat tvorbu vzorových příkladů, cvičení a testovacích úkolů, které pomohou studentům lépe porozumět a prakticky využívat Cosmos DB ve svých budoucích projektech.

Multimodelová schopnost Cosmos DB umožňuje vývojářům používat různé datové modely podle potřeb jejich aplikací, což činí tuto platformu extrémně flexibilní. V průběhu této práce budou probrány klíčové koncepty jako globální distribuce, konzistence dat, modely konzistence a strategie škálování, které jsou zásadní pro optimalizaci výkonu a efektivity databázových operací v cloudovém prostředí.

Nakonec, tato práce hodlá navrhnout také možné směry rozvoje učební pomůcky, což by mohlo zahrnovat integraci s jinými službami Azure, práci s reálnými datovými sadami nebo rozšíření o nové moduly zahrnující pokročilé techniky databázového managementu. Předpokládá se, že výsledky práce pomohou studentům technických oborů získat hlubší porozumění cloudovým databázovým technologiím a jejich využití v praxi.

Na oficiálním blogu Microsoft Azure Scott Guthrie zdůraznil [1] význam Azure Cosmos DB, zvláště v kontextu její schopnosti zpracovávat data na planetární úrovni. Uvedl, že Azure Cosmos DB je první globálně distribuovaná multimodelová databáze, která zajišťuje zpracování dat s garantovanou odezvou v milisekundách a vysokou dostupností. Tyto schopnosti, podle Guthrieho, otevírají nové možnosti pro vývojáře při vytváření inteligentních aplikací, zvláště těch, které se škálují pro globální uživatele a integrují se s technologiemi umělé inteligence.

## **I. TEORETICKÁ ČÁST**

## 1 SEZNÁMENÍ S COSMOS DB

Kapitola "Seznámení s Azure Cosmos DB" poskytuje detailní pohled na architekturu, klíčové vlastnosti a další informace týkající se práce s touto multimodelovou databází. Začíná přehledem architektury Cosmos DB a různých modelů dat, které tato platforma podporuje, včetně jejich využití a výhod. Dále se zabývá klíčovými koncepty, jako jsou distribuovaná konzistence, škálovatelnost a globální replikace dat, a vysvětluje, jak tyto prvky ovlivňují návrh a provoz aplikací. Tato kapitola slouží jako základní kámen pro pochopení fungování Azure Cosmos DB a poskytuje potřebné informace pro další prozkoumávání této databázové služby.

### 1.1 Historie vývoje Cosmos DB

Azure Cosmos DB má své kořeny v projektu, který byl zahájen v roce 2010 pod názvem "Project Florence". Tento projekt byl odpovědí na potřeby vývojářů uvnitř společnosti Microsoft, kteří čelili problémům při vývoji aplikací na velkou škálu. Když se ukázalo, že výzvy spojené s vývojem globálně distribuovaných aplikací nejsou unikátní jen pro Microsoft, rozhodli se v roce 2015 učinit tuto technologii dostupnou vývojářům Azure pod názvem Azure DocumentDB. Od té doby byla technologie neustále rozvíjena, přidávány byly nové funkce a zaváděny významné nové schopnosti. Výsledkem těchto snah je Azure Cosmos DB, která představuje další velký skok vpřed v oblasti globálně distribuovaných databází na cloudové platformě. S příchodem Azure Cosmos DB se zákazníci DocumentDB automaticky stali zákazníky Azure Cosmos DB, přičemž jejich data byla bezproblémově převedena a získali přístup k novým možnostem a vylepšením, které Azure Cosmos DB nabízí. [2]

### 1.2 Služby Azure od Microsoftu

Microsoft Azure poskytuje širokou škálu cloudových služeb, které umožňují vývojářům, IT specialistům a firmám vytvářet, nasazovat a spravovat aplikace prostřednictvím globální sítě datových center. Mezi hlavní kategorie služeb patří výpočetní služby, databázové služby, síťové služby, úložiště dat a služby pro identitu a přístupové řízení.

V kategorii výpočetních služeb najdeme Azure Virtual Machines, což jsou virtuální stroje pro nasazení a správu aplikací, Azure Kubernetes Service (AKS), který je spravovanou službou Kubernetes usnadňující nasazení a správu kontejnerizovaných aplikací, a Azure

Functions, platformu pro serverless výpočty umožňující spouštění kódu v reakci na události bez potřeby spravovat infrastrukturu.

Databázové služby zahrnují Azure Cosmos DB, globálně distribuovanou multimodelovou databázi optimalizovanou pro rychlé dotazy a flexibilní správu dat, Azure SQL Database, spravovanou relační databázovou službu zjednodušující správu a škálování aplikací, a Azure Database for MySQL a PostgreSQL, které jsou spravovanými službami poskytujícími plně spravované komunitní databáze.

Síťové služby zahrnují Azure Virtual Network pro vytváření izolovaných sítí v Azure, Azure ExpressRoute, který poskytuje privátní připojení z korporátní sítě do Azure, a Azure DNS pro hostování oblastí DNS, což zlepšuje výkon a dostupnost aplikací.

V oblasti úložiště dat máme Azure Blob Storage pro ukládání objemných binárních dat, Azure File Storage nabízející sdílené úložiště pro přístup k souborům a Azure Queue Storage, který poskytuje fronty zpráv pro komunikaci mezi komponentami aplikace.

Služby pro identitu a přístupové řízení zahrnují Azure Active Directory, spravovanou službu identity umožňující jednotné přihlašování a správu přístupu k aplikacím, a Azure Key Vault, který spravuje a ochraňuje kryptografické klíče a jiné tajemství používané aplikacemi a službami.

Azure nabízí mnoho dalších specializovaných nástrojů a služeb, které mohou pokrýt specifické potřeby různých oblastí IT a podnikání. Azure Cosmos DB je zvláště významná pro vývoj aplikací, které vyžadují globální distribuci a vysoký výkon při práci s daty. [3]

### 1.3 Architektura a klíčové vlastnosti

Microsoft Azure Cosmos DB je vysoce výkonná, globálně distribuovaná multimodelová databáze navržená tak, aby poskytovala extrémní škálovatelnost, vysokou dostupnost a flexibilní konzistenci dat přes různé geografické regiony. Díky své architektuře a klíčovým vlastnostem je Cosmos DB ideální pro podnikové aplikace, které potřebují spolehlivě zpracovávat velké objemy dat s minimální latencí, a to i v globálním měřítku. [4]

#### Hlavní aspekty architektury a vlastnosti Cosmos DB:

- 1) Globální distribuce a multi-regionální podpora: Cosmos DB umožňuje aplikacím transparentně a dynamicky přidávat nebo odebírat regiony, což zajišťuje bezproblémový přístup k datům a vysokou dostupnost. Tato funkcionality je klíčová

pro mezinárodní společnosti, které potřebují zajistit rychlý přístup k datům pro uživatele na různých kontinentech. [4]

- 2) Podpora různých modelů dat: Databáze podporuje různé datové modely, včetně dokumentů, klíč-hodnota, sloupcové rodiny a grafy. Tato multimodelová flexibilita umožňuje vývojářům vybrat nejvhodnější model pro každý konkrétní případ použití, aniž by bylo nutné spravovat více databázových technologií. [4]
- 3) Automatické škálování: Cosmos DB poskytuje možnost automatického škálování prostředků, včetně průchodnosti a úložiště, na základě aktuálního využití. Toto škálování je možné provádět jak vertikálně (zvětšení kapacity jednotlivých jednotek), tak horizontálně (přidávání více jednotek). [4]
- 4) Konzistence na míru: Nabízí pět úrovní konzistence - od silné po nakonec konzistentní. Tato flexibilita v nastavení konzistence umožňuje uživatelům optimalizovat přístup k datům v závislosti na požadavcích konkrétní aplikace na konzistenci a výkon. [4]
- 5) Zabezpečení a ochrana dat: Cosmos DB zahrnuje pokročilé zabezpečení, jako je šifrování dat v klidu a šifrování přenášených dat, správa přístupových práv a integrace s Azure Active Directory pro správu identit a přístupových oprávnění. Tato bezpečnostní opatření zajišťují, že data jsou chráněna jak na úrovni ukládání, tak při přenosu. [4]
- 6) Integrace s dalšími službami Azure: Cosmos DB je těsně integrována s řadou dalších služeb Azure, včetně Azure Functions pro serverless computing, Azure Machine Learning pro pokročilé analytické zpracování dat a Azure Cognitive Services pro vytváření inteligentních aplikací. [4]

Díky těmto vlastnostem je Azure Cosmos DB vhodná pro širokou škálu aplikací, od mobilních a webových aplikací po IoT (internet věcí) a globálně distribuované aplikace, které vyžadují rychlý a spolehlivý přístup k datům bez ohledu na to, kde se uživatelé nacházejí.

## 1.4 Modely dat a jejich použití

Azure Cosmos DB podporuje několik datových modelů, což vývojářům umožňuje zvolit nejvhodnější způsob ukládání a manipulace s daty pro specifické požadavky jejich aplikací. Tady je pohled na čtyři hlavní datové modely podporované v Cosmos DB a příklady, kde se obvykle používají.



#### 1.4.1 Dokumentově orientovaný model (NoSQL API)

Tento model ukládá data ve formátu JSON, což umožňuje velmi flexibilní schéma. Je ideální pro aplikace, které potřebují rychlý přístup k datovým položkám na základě klíčů. Dokumentově orientovaný model je často využíván webovými a mobilními aplikacemi, kde je třeba efektivně ukládat, číst a upravovat data, jako jsou uživatelské profily, objednávky e-shopů a další dynamické informace. [5]

#### 1.4.2 Grafový model (Gremlin API)

Umožňuje reprezentaci dat jako grafů skládajících se z uzlů (entity) a hran (vztahy). Tento model je vhodný pro aplikace, které zpracovávají komplexní vztahy mezi daty, jako jsou sociální sítě, systémy doporučení a analýzy síťových interakcí. [5]

#### 1.4.3 Model klíč-hodnota (Table API)

Tento model je optimalizován pro rychlé vyhledávání dat na základě klíče a je vhodný pro aplikace, které vyžadují velmi rychlý přístup k velkému množství jednoduchých datových položek. Typické použití zahrnuje webové session management, konfigurace aplikací a caching. [5]

#### 1.4.4 Sloupcově orientovaný model (Cassandra API)

Speciálně navržený pro správu velkých množství dat s nízkou latencí při čtení a zápisu, což je ideální pro big data aplikace a analýzu velkých datasetů v reálném čase, jako jsou telemetrická data a monitoring.

Každý z těchto modelů nabízí specifické výhody, které umožňují vývojářům stavět vysoce výkonné a škálovatelné aplikace, zatímco Cosmos DB zajišťuje, že všechny tyto různé typy dat jsou spravovány s vysokou dostupností a konzistencí na globálním měřítku. [5]

### 1.5 Klíčové koncepty

Azure Cosmos DB je vysoce škálovatelná a plně spravovaná NoSQL databázová služba od Microsoftu, navržená pro globální distribuci a reálné použití v mnoha různých aplikacích. Několik klíčových konceptů je základem pro její design a funkcionalitu, což zahrnuje globální distribuci, podporu více modelů dat, elastické škálování a komplexní konzistenci.

Prvně, globální distribuce v Cosmos DB umožňuje automatickou replikaci dat mezi geograficky rozloženými datacentry, což zajišťuje vysokou dostupnost a rychlý přístup k

datům pro uživatele bez ohledu na jejich lokaci. To je ideální pro aplikace vyžadující rychlé odezvy a vysokou dostupnost, jako jsou webové a mobilní aplikace, distribuované aplikace a mikroslužby.

Dále, Cosmos DB nabízí podporu pro několik datových modelů, včetně dokumentových, grafových, klíč-hodnota a sloupcových modelů. Toto umožňuje vývojářům vybrat si datový model, který nejlépe vyhovuje specifickým potřebám jejich aplikací. Například, dokumentový model je vhodný pro JSON-like struktury, zatímco grafový model je ideální pro aplikace, které zpracovávají složité vztahy mezi daty.

Elastické škálování je dalším důležitým prvkem, který umožňuje Cosmos DB automaticky škálovat výkon a úložný prostor dle potřeby. To zahrnuje možnost vertikálního i horizontálního škálování, což umožňuje aplikacím snadno se přizpůsobit měnícím se požadavkům na zatížení bez nutnosti manuálních zásahů.

Azure Cosmos DB nabízí pět úrovní konzistence, které uživatelům umožňují vybrat si úroveň konzistence podle specifických potřeb jejich aplikací. Silná konzistence je vhodná pro kritické aplikace jako finanční transakce, kde je nutná absolutní přesnost dat. Omezená stálá konzistence je ideální pro monitoring systémů, kde je přijatelné malé zpoždění dat. Relativní konzistence je perfektní pro uživatelské aplikace jako nákupní košíky, kde musí uživatelé vidět své změny okamžitě. Relativní konzistence (Consistent Prefix) je vhodná pro aplikace, kde je důležité, aby data byla zpracována ve správném pořadí, například pro zpracování logů. Eventuální konzistence je nejlepší pro aplikace, kde je prioritou vysoká dostupnost a výkon, například pro systémy s doporučeními nebo sociální sítě.

Tyto koncepty společně definují Azure Cosmos DB jako vysoce adaptabilní, spolehlivou a výkonnou databázovou platformu, která je schopna podporovat širokou škálu podnikových a globálních aplikací. [4]

## 2 STRATEGIE EFEKTIVNÍHO PLÁNOVÁNÍ VÝUKY S DŮRAZEM NA COSMOS DB

V dnešním rychle se vyvíjejícím technologickém světě je zásadní připravit studenty na efektivní a efektivní využívání nových technologií. Azure Cosmos DB, jako přední globálně distribuovaná multimodelová databázová služba od Microsoftu, představuje klíčovou platformu, kterou by měli ovládat budoucí IT profesionálové. Výuka těchto pokročilých technologií vyžaduje pečlivé a strategické plánování, které zajistí, že studenti nejenže pochopí technické aspekty databáze, ale také získají praktické dovednosti potřebné pro jejich budoucí kariéru.

Podle Johna Hattieho a Gregoryho Donoghue, ve své studii "Learning Strategies: A Synthesis and Conceptual Model" [6] zveřejněné ve Science of Learning, je zásadní zaměřit vzdělávací proces na strategie, které jsou základem pro hluboké porozumění a trvalé učení. Autoři identifikují, že efektivní výukové strategie, které spojují teoretické poznatky s praktickými dovednostmi, vedou k významnému zlepšení výkonu studentů.

S tímto vědomím je cílem této kapitoly poskytnout komplexní přehled o tom, jak můžeme strukturovat a implementovat výukový plán, který maximálně podporuje pochopení a praktické využití Azure Cosmos DB. Strategie, které zde budou představeny, se opírají o aktuální pedagogický výzkum a osvědčené postupy, a jsou zaměřeny na zajištění, že každý student může účinně přijímat a aplikovat znalosti v praxi.

### 2.1 Základy plánování výuky IT a Cosmos DB

V oblasti vzdělávání v informačních technologiích, zejména při začleňování pokročilých databázových systémů jako je Azure Cosmos DB, je zásadní, aby plánování kurzu bylo pečlivě promyšlené a strukturované. Efektivní pedagogické strategie, které kombinují teorii s praktickou aplikací, jsou klíčem k úspěchu ve výuce.

Na základě osvědčených metod uvedených v knize Kennetha D. Moorea, "Effective Instructional Strategies: From Theory to Practice" [7], a Richarda M. Feldera a Rebecy Brent, "Teaching and Learning STEM: A Practical Guide" [8], můžeme identifikovat několik kroků pro efektivní výuku IT, které jsou zvláště relevantní pro technologie jako je Cosmos DB.

Prvním krokem je pečlivé stanovení vzdělávacích cílů. Je důležité, aby tyto cíle byly specifické, měřitelné, dosažitelné, relevantní a časově omezené (SMART). V případě kurzu

zaměřeného na Cosmos DB by měly cíle zahrnovat nejen pochopení základních principů databázových operací, ale také praktické dovednosti v konfiguraci a optimalizaci databázových operací v reálném prostředí Azure.

Dále je důležité provést analýzu potřeb studentů. Tento krok pomáhá přizpůsobit výuku tak, aby odpovídala předchozím znalostem a dovednostem studentů v oblasti IT a databází. V případě Cosmos DB je zásadní zjistit, zda studenti již mají zkušenosti s jinými NoSQL databázemi nebo s principy cloud computing, což může ovlivnit hloubku a tempo kurzu.

Výběr obsahu a jeho organizace jsou rovněž klíčové. Vzdělávací materiály by měly pokrývat široké spektrum témat od základních konceptů Cosmos DB, jako jsou globální distribuce a multimodelový přístup, až po pokročilé techniky, jako jsou strategie indexace a optimalizace dotazů.

Pedagogické metody by měly být různorodé, aby podpořily angažovanost a aktivní učení. Praktické laboratoře, studie případů a skupinové projekty jsou ideální pro hlubší porozumění materiálu a rozvoj technických dovedností. V kontextu Azure Cosmos DB je praktické cvičení neocenitelné, protože umožňuje studentům experimentovat s nastavením a správou databáze v cloudovém prostředí.

Hodnocení a zpětná vazba by měly být integrovány průběžně během kurzu, aby se zajistilo, že vzdělávací cíle jsou splněny a studenti efektivně absorbují učivo. To může zahrnovat pravidelné kvízy, projektové prezentace a peer review.

Tento přístup k plánování a realizaci kurzu v oblasti Azure Cosmos DB a IT všeobecně podporuje nejen získání hlubokých znalostí, ale také rozvoj praktických dovedností, které jsou klíčové pro úspěch v technologicky orientovaném pracovním prostředí. [7,8]

## 2.2 Stanovení vzdělávacích cílů

Určení cílů pro výuku Azure Cosmos DB představuje základní kámen vytváření efektivního učebního plánu, který umožní studentům úspěšně osvojit si klíčové dovednosti spojené s touto technologií. S použitím metody SMART lze tyto cíle formulovat následovně [9]:

- 1) Specifický cíl: Studenti budou schopni vytvořit a spravovat databáze v Azure Cosmos DB a provádět CRUD operace (Create, Read, Update, Delete) na datových položkách.
- 2) Měřitelný cíl: Úspěšnost dosažení cíle bude vyhodnocena na základě schopnosti studentů demonstrujících provádění CRUD operací u zápočtu.

- 3) Dosažitelný cíl: Každý student bude dostávat podrobné instrukce a podporu ke každé fázi výuky, aby mohl postupovat a rozvíjet své dovednosti v souladu se svými schopnostmi a tempem učení.
- 4) Relevantní cíl: Získání dovedností v práci s Cosmos DB je nezbytné pro současné i budoucí pracovní výzvy v oblasti informačních technologií, a proto se cíle kurzu přímo zaměřují na aplikovatelnost těchto dovedností v praxi.
- 5) Časově omezený cíl: Studenti by měli být schopni provádět CRUD operace v Azure Cosmos DB do konce semestru, což umožní dostatečný čas pro postupné učení a praktickou aplikaci naučeného před finálním hodnocením.

## 2.3 Výběr obsahu a struktura materiálu

Při výběru obsahu a struktura materiálu pro kurz o Azure Cosmos DB je zásadní zajistit, že obsah bude systematicky pokrývat všechny klíčové aspekty této multimodelové databáze. Cílem je, aby studenti získali nejen teoretické znalosti, ale také praktické dovednosti, které mohou využít v reálných aplikacích. Podle Kennetha D. Moorea [7] je efektivní struktura materiálu základem pro úspěšné vzdělávací strategie, které kombinují teorii s praktickou aplikací.

Struktura obrazovacího materiálu by měla být navržena tak, aby maximálně podporovala učení a zapojení studentů. Při plánování výukového programu, zejména technických předmětů jako je Azure Cosmos DB, je důležité, aby struktura materiálu byla systematická, přehledná a interaktivní. Klíčové aspekty, které by měly být zohledněny při strukturování studijního materiálu.

### 2.3.1 Modularita

Obrazovací materiál by měl být rozdělen do logicky organizovaných modulů nebo kapitol. Každý modul by měl pokrývat specifický koncept nebo téma a měl by být relativně samostatný, což umožňuje studentům flexibilní přístup k učení ve vlastním tempu. [10]

### 2.3.2 Sekvencování

Informace by měly být prezentovány v logickém a postupném pořadí od jednoduchého k složitějšímu. To znamená začít základními koncepty a postupně přecházet k pokročilým tématům, jakmile studenti získají potřebné předpoklady. [10]

### 2.3.3 Integrace vizuálních pomůcek

Využití diagramů, grafů, videí a dalších vizuálních pomůcek může pomoci objasnit složité koncepty a zlepšit zapamatování. Vizuální materiály jsou obzvláště užitečné v technických a vědeckých oborech, kde mohou komplexní informace být náročné na pochopení. [10]

### 2.3.4 Interaktivní komponenty

Zahrnutí interaktivních prvků, jako jsou kvízy, interaktivní simulace a praktické cvičení, může zvýšit zapojení studentů a umožnit jim testovat své znalosti v praxi. Tyto komponenty také podporují aktivní učení, což je klíčové pro hlubší porozumění a dlouhodobou retenci informací. [10]

### 2.3.5 Zpětná vazba a hodnocení

Zajištění pravidelné zpětné vazby prostřednictvím samoocenění, peer reviews nebo instruktorem řízeného hodnocení pomáhá studentům identifikovat oblasti, které potřebují další zlepšení. Hodnocení by mělo být konstruktivní a motivující, s jasnými směrnicemi, jak dosáhnout zlepšení. [10]

### 2.3.6 Doprovodné materiály

Poskytnutí doplňkových zdrojů, jako jsou články, odkazy na další čtení a online zdroje, může studentům pomoci rozšířit jejich znalosti a poskytnout jim další kontext pro studované téma. [10]

### 2.3.7 Přístupnost a adaptabilita

Materiály by měly být přístupné pro všechny studenty, včetně těch se speciálními vzdělávacími potřebami. Použití univerzálního designu pro učení (UDL) může zajistit, že vzdělávací obsah je přístupný a adaptabilní pro různé styly učení a schopnosti.

Tato struktura pomáhá zajistit, že vzdělávací materiály jsou nejen efektivní ve výuce konceptů, ale také podporují zapojení a úspěch studentů ve vzdělávacím procesu. [10]

## 2.4 Pedagogické metody a techniky výuky IT

V oblasti výuky informačních technologií je klíčové využívat moderní pedagogické metody, které nejen podporují zapojení a interaktivitu studentů, ale také jim pomáhají rozvíjet praktické dovednosti a hluboké porozumění technologickým konceptům. Tyto metody by

měly odpovídat dynamice a rychlému vývoji IT sektoru. Některé z nejúčinnějších pedagogických přístupů pro výuku IT, včetně nových technologií jako je Azure Cosmos DB:

#### **2.4.1 Projektově Orientované Učení (Project-Based Learning, PBL)**

Tento přístup zapojuje studenty do praktických projektů, které simulují reálné technologické problémy a vyžadují aplikaci teoretických znalostí. PBL podporuje rozvoj kritického myšlení a problémového řešení a je ideální pro IT vzdělávání, kde je praktická zkušenost neocenitelná. [11]

#### **2.4.2 Flipped Classroom**

Model "převrácené třídy" optimalizuje třídní interakci a praktické aktivity tím, že studenti studují teoretický materiál mimo třídu, například prostřednictvím online přednášek, a třídní čas je věnován diskusím, cvičením a aplikaci učiva. Tento model je zvláště užitečný pro složité IT koncepty, které vyžadují hlubší porozumění. [12]

#### **2.4.3 Kolaborativní Učení**

Tato technika zahrnuje práci ve skupinách, kde studenti společně řeší úkoly a projekty. IT projekty často vyžadují týmovou spolupráci, a kolaborativní učení tak připravuje studenty na budoucí pracovní prostředí, zlepšuje komunikační dovednosti a podporuje sdílení znalostí. [13]

#### **2.4.4 Online Tutoriály a MOOCs**

Masivní otevřené online kurzy (MOOCs) a interaktivní online tutoriály nabízejí flexibilní a přístupnou cestu k IT vzdělání, která umožňuje studentům učit se ve vlastním tempu a podle svých časových možností. Tyto zdroje jsou často aktualizovány, což je klíčové v rychle se měnícím oboru jako je IT. [14]

#### **2.4.5 Adaptivní Učení**

Technologie adaptivního učení používá algoritmy pro analýzu výkonu studentů a přizpůsobení učebního materiálu jejich individuálním potřebám a tempo učení. Tento přístup je obzvláště užitečný v IT vzdělávání, kde může rozsah znalostí studentů výrazně kolísat. [15]

## 2.5 Hodnocení a zpětná vazba

Hodnocení a zpětná vazba jsou zásadní součástí vzdělávacího procesu, zejména v oblasti informačních technologií, kde rychlý rozvoj a aplikace nových technologií vyžadují neustálou aktualizaci a zlepšování dovedností. Efektivní hodnocení a kvalitní zpětná vazba pomáhají studentům lépe pochopit, co ovládají dobře a na čem potřebují pracovat. Dál probereme některé klíčové aspekty hodnocení a poskytování zpětné vazby ve vzdělávacích programech IT. [16]

### 2.5.1 Formy hodnocení

- 1) Formátivní hodnocení: Toto průběžné hodnocení se provádí během vzdělávacího procesu a je zaměřeno na zlepšování učení studentů. Může zahrnovat kvízy, průběžné projekty, recenze kódu, nebo peer review. Cílem je poskytnout studentům zpětnou vazbu, která je včasná a umožňuje jim zlepšit své výkony během kurzu.
- 2) Sumativní hodnocení: Toto hodnocení se provádí na konci vzdělávacího období a slouží k posouzení, do jaké míry studenti dosáhli stanovených učebních cílů. Často zahrnuje závěrečné zkoušky, závěrečné projekty nebo portfolia. [16]

### 2.5.2 Techniky zpětné vazby

- 1) Okamžitá zpětná vazba: V digitálních učebních prostředích, jako jsou online kurzy, lze použít nástroje pro automatickou zpětnou vazbu, které studentům ihned ukazují, kde udělali chybu a jak ji mohou opravit. Tato rychlá zpětná vazba je obzvláště užitečná v technických disciplínách, kde je správné pochopení procesů klíčové.
- 2) Konstruktivní zpětná vazba: Efektivní zpětná vazba by měla být specifická, relevantní a konstruktivní. Měla by poskytnout jasnou orientaci, jak dosáhnout zlepšení. V IT vzdělávání je důležité, aby zpětná vazba obsahovala konkrétní příklady správného a špatného kódu, designových rozhodnutí nebo algoritmických řešení. [16]



## **II. PRAKTICKÁ ČÁST**

### 3 STRUKTUROVANÝ PLÁN PRO VÝUKU

V této kapitole se zaměříme na vývoj uceleného výukového programu pro multimodelovou databázi Azure Cosmos DB, která je jednou z klíčových technologií v cloudových službách Microsoft Azure. Cílem tohoto plánu je poskytnout studentům komplexní pochopení všech aspektů Cosmos DB, od základních principů až po pokročilé funkce, a to prostřednictvím teoretických lekcí a praktických cvičení.

Strukturovaný výukový plán je rozvržen do deseti týdnů a každý týden se zaměřuje na specifickou oblast Cosmos DB. Plán zahrnuje uvedení do technologie, práci s různými datovými modely, metody správy a bezpečnosti dat, integraci s dalšími službami Azure, a další. Takový přístup zajišťuje, že studenti získají nejen teoretické znalosti, ale i praktické dovednosti prostřednictvím cvičení a projektových úkolů.

Výuka bude postupovat od základních konceptů k detailnějším a složitějším tématům, což studentům umožní postupně si osvojit a prohloubit znalosti potřebné pro efektivní využívání Cosmos DB v reálných aplikacích. Tento plán nejen podpoří teoretické poznatky, ale také rozvíjí praktické dovednosti studentů, které jsou nezbytné pro práci v dynamickém prostředí cloudových technologií.

Výukový plán klade důraz na interaktivní a aplikované učení, což studentům umožní nejen se naučit teorii, ale také získat zkušenosti s praktickým řešením problémů, které mohou čelit jako budoucí IT profesionálové v oblasti cloudových databázových systémů. Celkově plán připravuje studenty, aby byli schopni nejen porozumět technologii Azure Cosmos DB, ale také aktivně ji využívat a implementovat ve svých budoucích kariérách.

#### 3.1 Přednášky

##### 3.1.1 Formát a obsah přednášek

Přednášky budou strukturovány tak, aby poskytly studentům ucelený pohled na klíčové aspekty Azure Cosmos DB. Každá přednáška bude trvat přibližně 50 minut a bude se konat jednou týdně. Během těchto setkání budou prezentovány teoretické informace prostřednictvím PowerPointových prezentací, doplněných o demonstrace a případové studie. Hlavní cílem je zprostředkovat studentům hluboké pochopení teorie doplněné o příklady z praxe.

### 3.1.2 Obsahová náplň přednášek

- 1) Úvod a základy Cosmos DB - Přehled funkcí, architektury a výhod používání Cosmos DB.
- 2) Datové modely a API - Podrobný pohled na různé datové modely (dokumentové, klíč-hodnota, grafový, sloupcový), které Cosmos DB podporuje, a jejich vhodné využití.
- 3) Konzistence a škálování - Vysvětlení konceptů konzistence dat a možností škálování v distribuovaném prostředí.
- 4) Bezpečnost a správa - Zásady bezpečnosti a správa přístupových práv k datům v Cosmos DB.
- 5) Integrace a real-time aplikace - Jak integrovat Cosmos DB s jinými službami Azure a příklady real-time aplikací využívajících Cosmos DB.

### 3.1.3 Cíle přednášek

Cílem přednášek je poskytnout studentům:

- 1) Základní znalosti o Cosmos DB, včetně porozumění jejímu umístění v ekosystému Microsoft Azure.
- 2) Praktické dovednosti pro modelování dat a práci s různými API.
- 3) Pochopení pokročilých témat jako jsou konzistence, škálování, a integrace s dalšími cloudovými službami.
- 4) Přípravu na praktické využití těchto znalostí ve svých budoucích projektech nebo pracovních rolích.

### 3.1.4 Získané znalosti

Po absolvování přednášek budou studenti schopni:

- 1) Definovat a vysvětlit klíčové komponenty a funkcionality Azure Cosmos DB.
- 2) Rozumět různým datovým modelům a vybírat vhodné modely pro konkrétní aplikace.
- 3) Navrhovat a implementovat databázová řešení s využitím Cosmos DB, která jsou škálovatelná, bezpečná a optimalizovaná pro výkon.

Přednášky budou kombinovat teorii s praktickými ukázkami a diskuzemi, což studentům umožní lépe pochopit a zapamatovat si učivo. Tento přístup také podporuje interaktivní učení a motivuje studenty k dalšímu zkoumání a experimentování s technologiemi.

## 3.2 Prezentací pro přednášky

### 3.2.1 1. týden: Úvod do Azure Cosmos DB

Prezentace poskytuje přehled kurzu a jeho cílů, které zahrnují poskytnutí komplexního pochopení Azure Cosmos DB, teoretických základů a praktických dovedností pro práci s touto technologií. Kurz je strukturován do deseti týdnů, kde každý týden obsahuje přednášku a praktické cvičení.











Dále se prezentace zaměřuje na vysvětlení, co je Azure Cosmos DB, její klíčové vlastnosti a výhody. Jedná se o globálně distribuovanou multimodelovou databázovou službu navrženou pro vysokou dostupnost, škálovatelnost a nízkou latenci. Umožňuje práci s různými typy datových modelů, jako jsou dokumenty, klíč-hodnota, grafy a sloupce.

Následně je představena architektura Azure Cosmos DB, která zahrnuje regiony, účty, databáze, kontejnery a položky. Důraz je kladen na koncepty konzistence, oddílů a indexace, které jsou klíčové pro optimalizaci výkonu a efektivitu databázových operací.

Na konci prezentace jsou uvedeny doporučené literatury a zdroje pro další studium. Studenti by po této úvodní lekci měli mít základní pochopení Azure Cosmos DB a být připraveni na hlubší zkoumání této technologie v následujících týdnech.

**Why Azure Cosmos DB**

Supports "all" popular NoSQL databases as managed DBaaS that provides "limitless" scalability, low latency and 5-nines availability with financially backed SLAs

<p>✓</p> <p>Support all popular NoSQL databases</p> <p>NoSQL     </p> <p>MongoDB </p> <p>Native and open API support for MongoDB, Cassandra, Gremlin, PostgreSQL data and models</p>	<p>✓</p> <p>Guaranteed speed and &lt;10 ms latency anywhere</p> <p></p> <p>Guaranteed &lt;10 ms latency backed by SLA. Automatic scaling with instant limitless elasticity.</p>	<p>✓</p> <p>99.999% availability with zero RPO and RTO</p> <p></p> <p>Guaranteed 99.999% availability, zero RPO &amp; RTO, backed by SLA and enterprise-level security.</p>	<p>✓</p> <p>Instant scalability and fully managed DBaaS</p> <p></p> <p>Auto-sharding, auto-scaling and auto-indexing for blazing fast queries and low TCO</p>	<p>✓</p> <p>Integration with the rest of the Azure portfolio</p> <p></p> <p>Azure Functions, Azure Synapse Link (HTAP) etc.</p>
--	--	--	--	--

Univerzita Tomáše Bati ve Zlíně

Obrázek 1. Příklad slajdu z prezentaci. Proč Azure Cosmos DB?

### 3.2.2 2. týden: Datové modely a API

Prezentace "Datové modely a API v Azure Cosmos DB" se zaměřuje na různé datové modely podporované Azure Cosmos DB a poskytuje přehled jejich použití a výhod. Začíná vysvětlením, co je API (Application Programming Interface), které umožňuje různým softwarovým aplikacím vzájemnou komunikaci.

Prezentace pokračuje přehledem datových modelů, které Azure Cosmos DB podporuje. Tyto modely zahrnují dokumentově orientovaný model (SQL API), grafový model (Gremlin API), model klíč-hodnota (Table API) a sloupcově orientovaný model (Cassandra API). Každý model je podrobně popsán, včetně jeho použití, výhod a typických scénářů, kde je nejefektivnější.

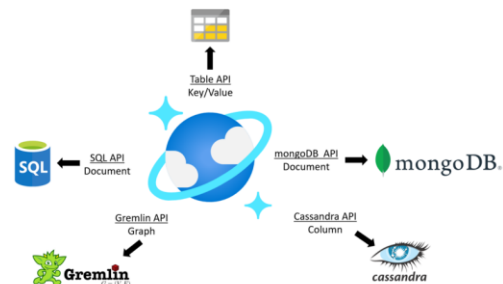
Dokumentově orientovaný model umožňuje práci s JSON dokumenty, což je flexibilní způsob ukládání různorodých dat. Grafový model je optimalizován pro práci s grafovými databázemi, které ukládají a dotazují se na uzly a hrany grafu. Model klíč-hodnota je jednoduchý a rychlý, ideální pro aplikace, které vyžadují rychlé čtení a zápis velkého množství dat. Sloupcově orientovaný model je navržen pro vysokou dostupnost a škálovatelnost, což je ideální pro aplikace, které potřebují zpracovávat velké objemy dat v distribuovaném prostředí.

Prezentace také zdůrazňuje výhody multimodelového přístupu, který umožňuje vývojářům vybrat nejvhodnější datový model pro konkrétní část aplikace, což zvyšuje flexibilitu a efektivitu při návrhu aplikací. Dále je diskutována škálovatelnost, vysoká dostupnost a konzistence dat v Azure Cosmos DB.

Na závěr prezentace by studenti měli mít hlubší pochopení různých datových modelů podporovaných Azure Cosmos DB a jak je použít ve svých aplikacích. V následujících lekcích se budou podrobněji zabývat konkrétními praktickými aspekty práce s Cosmos DB, včetně implementace CRUD operací a optimalizace výkonu.

## Přehled datových modelů v Azure Cosmos DB

- Azure Cosmos DB podporuje více datových modelů, což poskytuje vývojářům flexibilitu při návrhu jejich aplikací.
- Každý datový model je optimalizován pro specifické typy aplikací a způsob práce s daty.
- Tento přístup umožňuje kombinovat různé typy datových modelů v rámci jedné databázové platformy, což usnadňuje správu a integraci různých datových struktur.
- Podporované datové modely: Dokumentově orientovaný model (SQL API), Grafový model (Gremlin API), Model klíč-hodnota (Table API), Sloupcově orientovaný model (Cassandra API).



Univerzita Tomáše Bati ve Zlíně

Obrázek 2. Příklad slajdu z prezentaci. Přehled datových modelů v Cosmos DB.

### 3.2.3 3. týden: Konzistence dat

Tato prezentace se soustředí na klíčový aspekt distribuovaných databázových systémů – konzistenci dat. Konzistence definuje záruky ohledně viditelnosti změn dat mezi různými replikami databáze a je důležitá pro zajištění správnosti a integrity dat.

Azure Cosmos DB nabízí pět úrovní konzistence: silnou konzistenci, omezenou stálou konzistenci, relativní konzistenci, konzistenci prefixu a eventuální konzistenci. Každá z těchto úrovní nabízí různé kompromisy mezi výkonem, dostupností a latencí. Prezentace podrobně popisuje každou z těchto úrovní, jejich použití, výhody a nevýhody.

Silná konzistence zaručuje nejvyšší úroveň spolehlivosti a správnosti dat, což je ideální pro kritické aplikace jako finanční transakce, avšak přichází s vyšší latencí a nižší dostupností. Omezená stálá konzistence zaručuje, že čtení nebudou zastaralá více než určitý počet verzí nebo časový interval, což je vhodné pro aplikace vyžadující rovnováhu mezi konzistencí a výkonem. Relativní konzistence zaručuje konzistenci v rámci jedné klientské relace, což je užitečné pro webové a mobilní aplikace. Konzistence prefixu zajišťuje, že žádné čtení nebude zobrazovat změny mimo pořadí, a eventuální konzistence zajišťuje, že všechny repliky budou nakonec konzistentní, což je vhodné pro aplikace tolerující dočasné nekonzistence.

Prezentace také zdůrazňuje, jak vybrat správný model konzistence na základě požadavků na konzistenci dat, výkonnostní nároky a dostupnost. Na závěr studenti získají hluboké pochopení různých modelů konzistence nabízených Azure Cosmos DB a jejich použití ve svých aplikacích.

## Modely konzistence v Azure Cosmos DB

- Azure Cosmos DB nabízí pět úrovní konzistence.

Úroveň konzistence	Konzistence	Dostupnost	Latence	Propustnost
Strong	Nejvyšší	Nízká	Vysoká	Nízká
Bounded Staleness	Vysoká	Nízká	Vysoká	Nízká
Session	Střední	Vysoká	Střední	Střední
Consistent Prefix	Nízká	Vysoká	Nízká	Střední
Eventual	Nízká	Vysoká	Nízká	Vysoká

Obrázek 3. Příklad slajdu z prezentaci. Modely konzistence v Azure Cosmos DB.

### 3.2.4 4. týden: Škálování a dostupnost

Tato prezentace je věnována škálování a dostupnosti, které jsou klíčovými aspekty moderních distribuovaných databázových systémů. Prezentace začíná vysvětlením důležitosti škálování a dostupnosti v kontextu Azure Cosmos DB a jak tyto vlastnosti umožňují aplikacím zůstat výkonné a dostupné i při rostoucím objemu dat a zátěži.

Horizontální a vertikální škálování jsou dva hlavní způsoby, jak rozšiřovat kapacitu systému. Horizontální škálování (scaling out) zahrnuje přidávání nebo odstraňování databázových uzlů, zatímco vertikální škálování (scaling up) zahrnuje přidávání nebo odstraňování prostředků, jako je CPU a RAM, na jednotlivých uzlech. Cosmos DB podporuje horizontální škálování automatickým rozdělováním dat do oddílů, což zvyšuje výkon a škálovatelnost.

Prezentace také popisuje nastavení propustnosti v Cosmos DB měřené v RU/s (Request Units per Second). Toto nastavení je klíčové pro dosažení optimálního výkonu a nákladů. Dynamické škálování umožňuje automatické úpravy RU/s na základě aktuální zátěže, zatímco statické škálování vyžaduje manuální nastavení a úpravy.

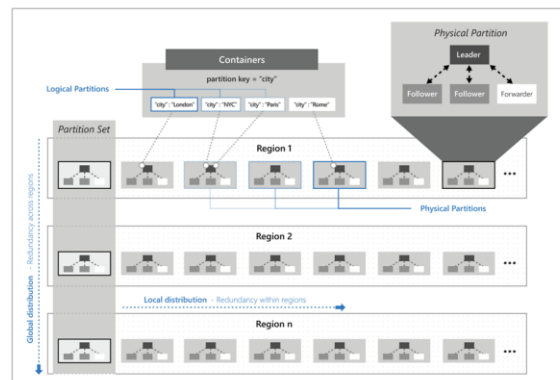
Dále se věnuje globální distribuci a replikaci dat, což zvyšuje dostupnost a snižuje latenci přístupu k datům. Cosmos DB umožňuje replikaci dat do více geografických regionů a podporuje automatické i manuální failovery, což zajišťuje kontinuitu provozu i při výpadcích.

Prezentace zahrnuje i SLA (Service Level Agreements) poskytované Microsoft Azure Cosmos DB, které garantují dostupnost až 99.999% pro víceregionální nasazení.

Na závěr jsou uvedeny praktické scénáře škálování a dostupnosti, jako jsou e-commerce aplikace během slevových akcí a globální aplikace sociálních médií, které využívají dynamické škálování a globální distribuci dat pro zajištění nízké latence a vysoké dostupnosti. Prezentace končí doporučeními pro optimalizaci výkonu a nákladů v Azure Cosmos DB.

## Oddíly v Azure Cosmos DB

- Data v Cosmos DB jsou rozdělena do oddílů pro dosažení lepšího výkonu a škálovatelnosti.
- Každý oddíl je identifikován klíčem oddílu, což je hodnota použitá pro rozdělení dat.
- Statické oddíly mají pevně danou velikost a nelze je měnit.
- Dynamické oddíly se automaticky přizpůsobují růstu dat a zátěži.



Univerzita Tomáše Bati ve Zlíně

Obrázek 4. Příklad slajdu z prezentaci. Oddíly v Azure Cosmos DB.

### 3.2.5 5. týden: Indexace a dotazování

Tato prezentace se zaměřuje na indexaci a dotazování v Azure Cosmos DB, klíčové aspekty pro zajištění rychlého a efektivního přístupu k datům. Začíná úvodem do indexace, kde se vysvětluje, jak indexace organizuje data tak, aby dotazy na databázi byly rychlejší a efektivnější. Indexace umožňuje databázovému systému rychle lokalizovat konkrétní data bez nutnosti prohledávat každý záznam, což výrazně zlepšuje výkon dotazů.



Prezentace pokračuje popisem principů indexace v Cosmos DB, která využívá invertovaný index k mapování hodnot zpět na jejich polohu v dokumentech. Tento přístup umožňuje rychlé vyhledávání a podporuje širokou škálu dotazů včetně těch, které obsahují filtry a agregace. Uživatelé mohou přizpůsobit indexační politiku podle potřeb svých aplikací.

Dále se prezentace věnuje dotazování v Cosmos DB. Podporuje dotazovací jazyk založený na SQL, který umožňuje vykonávat složité dotazy na JSON dokumenty. SQL API poskytuje bohatý výběr operátorů a funkcí pro práci s daty, což zahrnuje filtraci, řazení, agregaci a transformace dat. Efektivní dotazy jsou klíčem k vysokému výkonu aplikací, a optimalizace může zahrnovat úpravy dotazů pro minimalizaci počtu RU/s spotřebovaných během operací.

V další části se prezentace zabývá využitím indexů pro zrychlení dotazů. Indexy umožňují databázi vynechat nerelevantní data a zaměřit se pouze na záznamy odpovídající kritériím dotazu. Dobře navržené indexy mohou značně snížit latenci a zvýšit propustnost databázových operací. Prezentace doporučuje monitorovat výkon dotazů a upravovat indexační strategie podle potřeb pomocí nástrojů Azure Cosmos DB pro analýzu výkonu dotazů.

Na závěr prezentace jsou uvedeny příklady z praxe, kde e-commerce platformy používají indexy pro rychlé vyhledávání produktů na základě různých atributů a analytické aplikace využívají složité dotazy pro generování sestav a přehledů z velkých objemů dat. Diskutuje se také o běžných výzvách při indexaci a dotazování v Cosmos DB a o strategiích, jak tyto výzvy řešit.

Prezentace končí shrnutím klíčových bodů a zdůrazněním významu správného využití indexace a dotazovacích technik pro efektivní správu a provoz databázových systémů v Cosmos DB.

## Využití indexů pro zrychlení dotazů

- Indexy umožňují databázi vynechat nerelevantní data a zaměřit se pouze na záznamy odpovídající kritériím dotazu.
- Dobře navržené indexy mohou značně snížit latenci a zvýšit propustnost databázových operací.
- Monitorujte výkon vašich dotazů a upravujte indexační strategie podle potřeb.
- Využijte nástroje Azure Cosmos DB pro analýzu výkonu dotazů a identifikaci možností pro další optimalizaci.

 Univerzita Tomáše Bati ve Zlíně

Obrázek 5. Příklad slajdu z prezentaci. Využití indexů pro zrychlení dotazů.

### 3.2.6 6. týden: Bezpečnost a správa přístupu

Tato prezentace se soustředí na bezpečnost a správu přístupu v Azure Cosmos DB.

Nejprve se přehledně představí důležitost bezpečnosti dat a správy přístupu, zdůrazňující ochranu citlivých dat, zajištění souladu s regulačními předpisy, a zabezpečení integrity a dostupnosti dat. Poté se podrobně popisují bezpečnostní funkce Cosmos DB, jako je šifrování dat v klidu a při přenosu, správa šifrovacích klíčů pomocí Azure Key Vault a možnost uživatelů spravovat své vlastní klíče (Customer-Managed Keys).

Následuje část o správě přístupových práv pomocí Role-Based Access Control (RBAC) a integrace s Azure Active Directory (AAD), která poskytuje jednotné přihlášení, vícefaktorovou autentizaci a pokročilé zabezpečení. Poté se věnuje síťové bezpečnosti, včetně připojení přes virtuální síť (VNet), použití VNet Peering pro zvýšení bezpečnosti a snížení latence, a nastavení firewallu a přístupových seznamů (ACL).

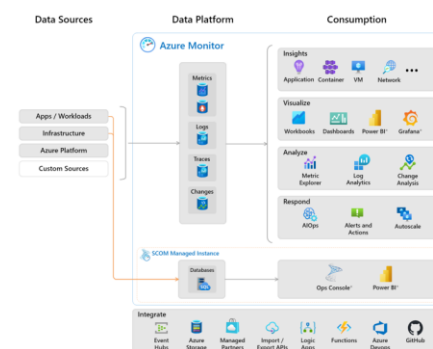
Další část se zaměřuje na auditování a sledování přístupu, kde se představují auditní logy, diagnostické protokoly, Azure Monitor a Azure Security Center. Tyto nástroje pomáhají identifikovat a řešit bezpečnostní incidenty a sledovat aktivity a přístup k databázi.

Prezentace dále popisuje, jak Cosmos DB podporuje soulad s předpisy a standardy, jako jsou GDPR, HIPAA, ISO/IEC 27001 a SOC 1, SOC 2 a SOC 3, což zajišťuje vysokou úroveň bezpečnosti a soukromí.

Nakonec jsou představeny nejlepší postupy pro zabezpečení Cosmos DB, včetně použití šifrování dat, pravidelné rotace šifrovacích klíčů, implementace RBAC, integrace s AAD, použití virtuálních sítí a firewallu, a pravidelného sledování auditních záznamů. Prezentace je zakončena shrnutím klíčových bodů a výhledem na další témata kurzu.

## Auditování a sledování přístupu

- Audit logy: Cosmos DB generuje auditní záznamy, které sledují aktivity a přístup k databázi.
- Azure Monitor: Umožňuje sledovat a analyzovat auditní záznamy, což pomáhá identifikovat a řešit bezpečnostní incidenty.



Univerzita Tomáše Bati ve Zlíně

Obrázek 6. Příklad slajdu z prezentaci. Auditování a sledování přístupu.

### 3.2.7 7. týden: Monitorování a ladění výkonu

Tato prezentace je věnována monitorování a ladění výkonu v Azure Cosmos DB.

Nejprve je zdůrazněna důležitost monitorování výkonu pro udržení optimálního provozu databázových aplikací, včetně identifikace úzkých míst, plánování škálování a zajištění, že aplikace splňuje požadavky na výkon a dostupnost. Dále se představují klíčové metriky výkonu, jako jsou Request Units per Second (RU/s), latence a propustnost, které pomáhají pochopit, jak efektivně databáze zpracovává požadavky a jaké zdroje jsou potřebné pro její běžný provoz.

Následuje část o ladění a optimalizaci výkonu, kde se probírají techniky, jako je optimalizace dotazů a správné indexování dat, které mohou značně zlepšit výkon databáze. Prezentace se také věnuje nástrojům pro monitorování a ladění, jako jsou Azure Monitor a Azure Cosmos DB Insights, které poskytují detailní přehledy o výkonu databáze a umožňují rychle reagovat na problémy.

Dále jsou uvedeny případové studie a konkrétní příklady z praxe, které ukazují, jak bylo díky monitorování a ladění dosaženo významného zlepšení výkonu. Diskutují se strategie, které byly použity k řešení specifických problémů výkonu.

Prezentace je zakončena shrnutím klíčových bodů a zdůrazněním důležitosti monitorování a ladění výkonu pro udržení optimálního stavu aplikace.

## Nástroje pro monitorování a ladění

- Azure Monitor: Poskytuje komplexní přehled o výkonu aplikací a infrastruktury.
- Azure Cosmos DB Insights: Specifický nástroj pro sledování výkonu databází Cosmos DB, který nabízí detailní pohledy na operace a zdroje.
- Pomocí těchto nástrojů můžete získat hlubší vhled do výkonu vaší databáze, identifikovat potenciální problémy a rychle reagovat na měnící se podmínky.



Univerzita Tomáše Bati ve Zlíně

Obrázek 7. Příklad slajdu z prezentaci. Nástroje pro monitorování a ladění.

### 3.2.8 8. týden: Integrace s dalšími Azure službami

Tato prezentace se zaměřuje na integraci Azure Cosmos DB s jinými službami Azure, což umožňuje vytvářet robustní, škálovatelné a vysoce dostupné aplikace.

Prezentace začíná přehledem integrace Azure služeb, kde je zdůrazněno, jak Cosmos DB může být dobře integrována s dalšími službami Azure. To zlepšuje výkon, bezpečnost a efektivitu aplikací, umožňuje automatizaci workflow, zlepšuje řízení datových toků a umožňuje hlubší analýzy a zpracování dat.

Následuje podrobné představení integrace s Azure Functions, což je bezserverová výpočetní služba, která umožňuje spouštět kód na základě událostí v cloudu. Cosmos DB může automaticky spouštět funkce v Azure Functions při přidání nebo změně dat.

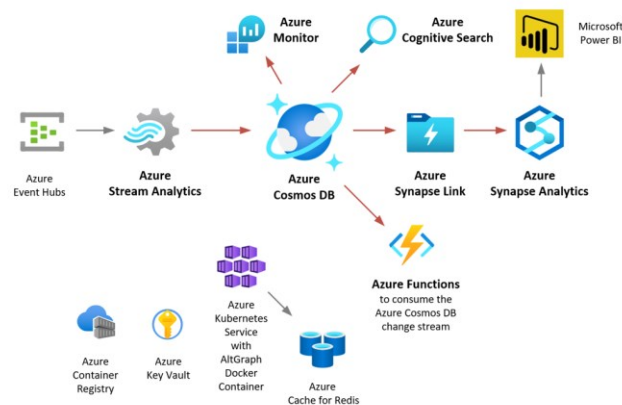
Další část se věnuje Azure Logic Apps, což je cloudová služba pro automatizaci a orchestraci úloh a pracovních postupů mezi různými aplikacemi a službami. Integrace s Cosmos DB umožňuje automatizaci procesů jako je zpracování objednávek a správa inventáře.

Prezentace také pokrývá integraci s Azure Machine Learning, což umožňuje použití dat z Cosmos DB pro trénování modelů strojového učení. To je užitečné pro automatizaci procesů jako je detekce podvodů a personalizace obsahu.

Další je integrace s Azure Stream Analytics, službou pro zpracování dat v reálném čase, která umožňuje analýzu dat proudících do nebo z Cosmos DB. To je ideální pro aplikace vyžadující okamžitou reakci na události.

Prezentace končí shrnutím, že integrace Azure Functions, Logic Apps, Machine Learning a Stream Analytics rozšiřuje možnosti Cosmos DB a umožňuje vývojářům vytvářet dynamické a škálovatelné aplikace, které efektivně využívají cloudové technologie.

## Přehled integrace Azure služeb



Obrázek 8. Příklad slajdu z prezentaci. Přehled integrace Azure služeb.

### 3.2.9 9. týden: Migrace dat do Cosmos DB

Tato prezentace se zaměřuje na migraci dat do Azure Cosmos DB, což je klíčový krok pro organizace, které chtějí využívat výhody cloudového úložiště. Prezentace začíná úvodem do migrace dat, kde se vysvětluje důležitost správného plánování a provedení migrace pro minimalizaci přerušení a zajištění integrity dat.

Dále se prezentace věnuje plánování migrace, kde se probírá analýza zdrojových dat, výběr strategie migrace (přímá, postupná nebo hybridní), zajištění kompatibility datových modelů a vliv migrace na běžící aplikace.

Prezentace také představuje nástroje pro migraci, jako jsou Azure Data Factory, která poskytuje vizuální nástroje pro tvorbu, plánování a monitorování migrací, a Cosmos DB Data Migration Tool, který je speciálně navržený pro import dat do Cosmos DB z různých zdrojů.

Nakonec se prezentace zaměřuje na shrnutí a doporučení, kde se zdůrazňuje důkladná příprava a analýza zdrojových dat, výběr správných nástrojů a strategií, kontinuální monitorování a optimalizace po migraci a plánování post-migračních úprav a optimalizací.

Cílem prezentace je poskytnout studentům solidní základ pro plánování a provádění migrace dat do Azure Cosmos DB a zdůraznit důležitost důkladného plánování a správného výběru nástrojů pro úspěšnou migraci.

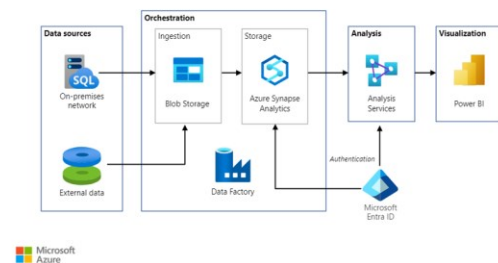
## Nástroje pro migraci

### Azure Data Factory

- Služba pro integraci dat, která podporuje kopírování dat mezi různými zdroji a cíli.
- Poskytuje vizuální nástroje pro tvorbu, plánování a monitorování migrací.

### Cosmos DB Data Migration Tool

- Nástroj od Microsoftu speciálně navržený pro import dat do Cosmos DB z různých zdrojů, včetně SQL databází, MongoDB, CSV souborů a dalších.



Obrázek 9. Příklad slajdu z prezentaci. Nástroje pro migraci.

### 3.2.10 10. týden: Case studies a best practices

Tato prezentace pojednává o případových studiích a osvědčených postupech pro Azure Cosmos DB, konkrétně s příkladem společnosti NewDay, jednoho z největších poskytovatelů spotřebitelského úvěru ve Velké Británii.

V úvodu se představuje NewDay jako inovativní společnost, která poskytuje nezajištěné úvěry širokému spektru spotřebitelů. Společnost obsluhuje téměř 5 milionů zákazníků a je známá svým moderním přístupem k poskytování kreditních produktů.

Následně se probírají výzvy a cíle NewDay, včetně omezení původní platformy na jednu instalaci, což komplikovalo škálování a globální expanzi. Cílem bylo zlepšit schopnost přímého zapojení značek do poskytování kreditních produktů zákazníkům a implementovat řešení umožňující okamžitou škálovatelnost, nízkou latenci a vysokou dostupnost.

Dále je popsáno řešení pomocí Azure Cosmos DB. NewDay zvolila tuto technologii pro její schopnosti v oblasti globální distribuce a škálovatelnosti. Azure Databricks byl použit pro hladkou migraci dat s nulovým výpadkem, což umožnilo NewDay experimentovat a rychle přidávat nové typy dat a domény, čímž podpořilo inovace ve společnosti.

Výsledky a dopady implementace Azure Cosmos DB jsou rovněž diskutovány. NewDay transformovala svůj obchodní model a rozšířila své služby na poskytování technologických platforem jiným finančním institucím. Došlo k výraznému zlepšení času odezvy a zvýšení spolehlivosti systému. NewDay nyní poskytuje end-to-end řešení pro finanční instituce, včetně technologií underwriting a dalších služeb.

Na závěr se prezentace věnuje osvědčeným postupům, kde se zdůrazňuje flexibilita a škálovatelnost Azure Cosmos DB, což umožnilo NewDay přizpůsobit se rychle měnícím tržním podmínkám. Důraz je kladen na plánování a spolupráci s technologickými partnery jako Microsoft pro optimalizaci výkonu a nákladů. Doporučuje se nebát se experimentovat s novými technologickými řešeními, aby firma mohla inovovat a zůstat konkurenceschopná, a využívat cloudové technologie pro zlepšení operací a zákaznické zkušenosti.

## Úvod do případové studie **NewDay**

- NewDay je jedním z největších poskytovatelů spotřebitelského úvěru ve Velké Británii, obsluhující téměř 5 milionů zákazníků.
- Společnost je známá svým inovativním přístupem k poskytování nezajištěných úvěrů široké škále spotřebitelů.

Obrázek 10. Příklad slajdu z prezentaci. Case Study společnosti NewDay.

### 3.3 Cvičení

V rámci cvičení k Azure Cosmos DB budeme kombinovat teoretické znalosti s praktickými dovednostmi. Cílem cvičení je umožnit studentům aplikovat teoretické koncepty na reálné příklady a projekty, což jim pomůže lépe pochopit funkce a možnosti této databáze.

#### 3.3.1 Popis průběhu cvičení

Každé cvičení bude trvat 50 minut a bude se konat jednou týdně, ideálně následující den po přednášce pro dané téma. Tímto způsobem budou studenti moci aplikovat a upevnit znalosti získané během přednášek.

#### 3.3.2 Obsah cvičení

- 1) Úvodní cvičení: Seznámení s rozhraním Azure Cosmos DB, základy nastavení databází a první dotazy. Studenti se naučí, jak vytvořit databázi a jak manipulovat s daty.
- 2) Práce s datovými modely: Studenti budou pracovat s různými datovými modely podporovanými Cosmos DB, včetně dokumentových a grafových modelů. Cílem je pochopení, kdy a jak efektivně využívat jednotlivé modely.



- 3) Škálování a konzistence: Zaměříme se na implementaci škálování a správu konzistence v Cosmos DB. Studenti budou experimentovat s různými úrovněmi konzistence a zjistí, jak tyto nastavení ovlivňují výkon a dostupnost dat.
- 4) CRUD operace:
  - Vložení (Create): Jak přidat nové dokumenty do databáze.
  - Čtení (Read): Dotazování a čtení dat z databáze.
  - Aktualizace (Update): Úprava existujících dokumentů.
  - Mazání (Delete): Odstranění dokumentů z databáze.

### 3.3.3 Zápočtový úkol

Na konci semestru bude pořádán zápočtový test, který ověří celkové pochopení a praktické dovednosti studentů. Test bude mít formu projektu, kde studenti vytvoří aplikaci využívající Azure Cosmos DB. Projekt bude hodnocen na základě technické správnosti, inovace a kvality implementace. Datum odevzdání projektu bude oznámeno v polovině semestru, maximální počet bodů bude 100, kde pro úspěšné absolvování kurzu bude potřeba získat minimálně 50 bodů.

Tento strukturovaný a prakticky orientovaný přístup zajišťuje, že studenti nejen získají hluboké teoretické znalosti o Azure Cosmos DB, ale také získají nezbytné dovednosti pro jejich budoucí kariéru v oblasti cloudových technologií.

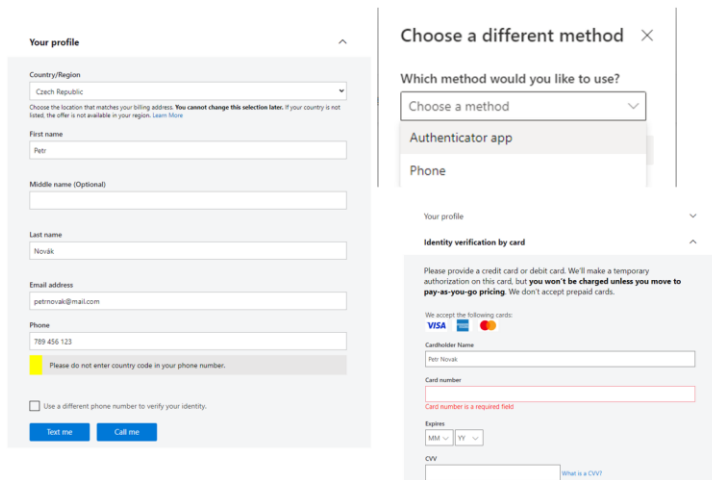
## 3.4 Prezentací pro praktické cvičení

### 3.4.1 1. týden: Úvod do prostředí Azure a vytvoření účtu Azure

Tato prezentace se zaměřuje na úvod do prostředí Azure a vytvoření účtu Azure. Azure je cloudová platforma od Microsoftu, která poskytuje širokou škálu služeb včetně výpočetních, analytických, úložných a síťových služeb. Umožňuje vytvářet, spravovat a nasazovat aplikace a služby prostřednictvím globální sítě datových center. Výhody Azure zahrnují flexibilitu, škálovatelnost, bezpečnost a globální dosah. Cílem této části kurzu je naučit studenty vytvářet účet Azure a orientovat se v Azure Portálu. Prezentace zahrnuje kroky pro vytvoření účtu, včetně zadání osobních údajů, kontaktních informací a platebních údajů pro ověření identity. Dále se studenti naučí používat Azure Portál, včetně hlavních prvků jako jsou Home, Dashboard, All resources, Resource groups a Marketplace. Vysvětluje se, jak přidávat nové zdroje a jak používat navigační menu a vyhledávací pole pro rychlé nalezení

konkrétních zdrojů nebo nastavení. Celkově tato prezentace poskytuje základní znalosti potřebné pro práci s Azure a vytváření a správu účtu v Azure Portálu.

## Vytvoření účtu Azure



The image shows two overlapping screenshots from the Azure portal's account creation process. The left screenshot, titled 'Your profile', shows a form with fields for Country/Region (Czech Republic), First name (Petr), Middle name (Optional), Last name (Novák), Email address (petrnovak@mail.com), and Phone (789 456 123). The right screenshot, titled 'Choose a different method', shows a dropdown menu for 'Choose a method' with 'Authenticator app' selected, and a 'Phone' field. Below this is a section for 'Identity verification by card' which includes a warning about temporary authorization, a list of accepted cards (VISA, Mastercard), and fields for Cardholder Name (Petr Novak), Card number (with a red error message 'Card number is a required field'), Expiry date (MM/YY), and CVV. A small logo for 'Univerzita Tomáše Bati ve Zlíně' is visible in the bottom right corner of the right screenshot.

Obrázek 11. Příklad slajdu z prezentaci. Vytvoření účtu Azure.

### 3.4.2 2. týden: Vytvoření účtu Cosmos DB

Tato prezentace je věnována vytvoření účtu Cosmos DB. Prezentace začíná úvodem do Cosmos DB, kde je vysvětleno, že Azure Cosmos DB je globálně distribuovaná multi-modelová databáze od Microsoftu, která poskytuje škálovatelnost a vysokou dostupnost. Cosmos DB podporuje různé datové modely a API včetně dokumentového modelu (SQL API), grafového modelu (Gremlin API), modelu klíč-hodnota (Table API) a sloupcově orientovaného modelu (Cassandra API). Následují kroky pro vytvoření nového účtu Cosmos DB, které zahrnují přihlášení do Azure Portálu, vyhledání služby Cosmos DB, výběr typu API, vyplnění potřebných informací jako je subscription, resource group, account name a region, a konečně vytvoření účtu. Po vytvoření účtu je představena hlavní sekce Cosmos DB účtu, jako jsou Overview, Data Explorer, Scale & Settings, Keys a Replicate Data Globally. Dále je vysvětleno, proč je Azure Cosmos DB for NoSQL doporučený typ databáze pro začátečníky díky své jednoduchosti, širokému použití a snadným CRUD operacím. Celkově tato prezentace poskytuje krok za krokem návod pro vytvoření a základní konfiguraci účtu Cosmos DB.

## Vytvoření účtu Cosmos DB

Home >  
Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Project Details  
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* Azure subscription 1  
Resource Group \* (New) myResourceGroup  
[Create new](#)

Instance Details  
Account Name \* petrovak

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones  Enable  Disable

Location \* (Europe) Poland Central  
Available locations are determined by your subscription's access and availability zone support (if that is enabled). If you don't see or cannot select your desired location, please open a support request for region access. [Click here for more details on how to create a region access request.](#)

Due to the location you have selected, a backup storage redundancy option must be selected. Please go to the [Backup policy](#) tab and select the desired backup storage redundancy option before you create this account.

Capacity mode  Provisioned throughput  Serverless  
[Learn more about capacity mode](#)

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated \$4/month discount per account.

Apply Free Tier Discount  Apply  Do not Apply

Limit total account throughput  Limit the total amount of throughput that can be provisioned on this account  
This limit will prevent unexpected charges related to provisioned throughput. You can update or remove this limit after your account is created.

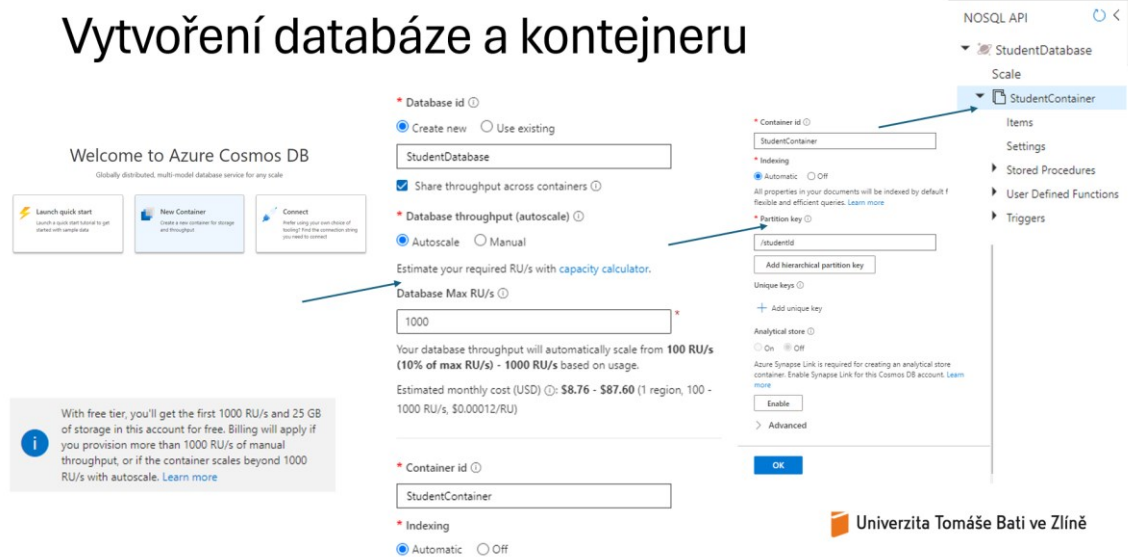
[Review + create](#) [Previous](#) [Next: Global Distribution](#)

Univerzita Tomáše Bati ve Zlíně

Obrázek 12. Příklad slajdu z prezentaci. Vytvoření účtu Cosmos DB.

### 3.4.3 3. týden: Úvod do základních konceptů a architektury Cosmos DB

Tato prezentace se zaměřuje na základní koncepty a architekturu Azure Cosmos DB. Úvodní část vysvětluje, co je Azure Cosmos DB, a seznamuje studenty s hlavními komponentami její architektury, jako jsou regiony, účty, databáze, kontejnery a položky. Následně se prezentace věnuje různým úrovním konzistence dat v Azure Cosmos DB, jako jsou silná konzistence, omezená stálost, relativní konzistence, konzistentní předpona a eventuální konzistence. Studenti se dále seznámí s pojmem oddílů (partitions) a jejich rolí ve škálovatelnosti databáze. Prezentace také pokrývá koncept indexace a její význam pro zrychlení dotazování. Praktická část prezentace obsahuje kroky pro vytvoření nové databáze a kontejneru v Azure Cosmos DB pomocí Azure Portálu, včetně nastavení partition key. Celkově tato prezentace poskytuje komplexní přehled o klíčových principech a technologiích, které tvoří základ Azure Cosmos DB.



Obrázek 13. Příklad slajdu z prezentaci. Vytvoření databáze a kontejneru.

### 3.4.4 4. týden: Škálování a dostupnost + zadání projektu

Tato prezentace se soustředí na problematiku škálování a dostupnosti v Azure Cosmos DB a zadání projektu. V úvodní části se studenti seznámí s koncepty škálování a dostupnosti, což jsou klíčové aspekty pro zajištění výkonu a spolehlivosti databáze. Škálování zahrnuje horizontální a vertikální metody, které umožňují rozšíření kapacity a výkonu podle potřeb aplikace. Horizontální škálování rozděluje data mezi více serverů, zatímco vertikální škálování zvyšuje kapacitu jednoho serveru.

Dostupnost je zajištěna prostřednictvím globální distribuce dat a automatického přepnutí v případě výpadků, což je podpořeno SLA garancí dostupnosti 99.999%. Studenti se naučí, jak konfigurovat škálování a regionální replikaci v Azure Cosmos DB prostřednictvím Azure Portálu. Tento proces zahrnuje nastavení kapacity (RU/s), výběr mezi Provisioned Throughput a Serverless, a přidávání nebo odstraňování regionů.

V závěrečné části je zadání projektu, kde studenti vytvoří jednoduchou databázi pro ukládání informací o knihách v knihovně, implementují základní CRUD operace a obhajují svůj projekt. Studenti musí dodat zdrojový kód, dokumentaci projektu a prezentaci projektu. Dokumentace by měla obsahovat úvod, přehled projektu, popis architektury databáze, implementované funkce, příklady použití CRUD operací, bezpečnostní opatření a závěry. Prezentace by měla být stručná, přehledná a vizuálně atraktivní.

Tato prezentace poskytuje komplexní přehled o škálování a dostupnosti v Azure Cosmos DB a připravuje studenty na praktickou implementaci a obhajobu jejich projektů.

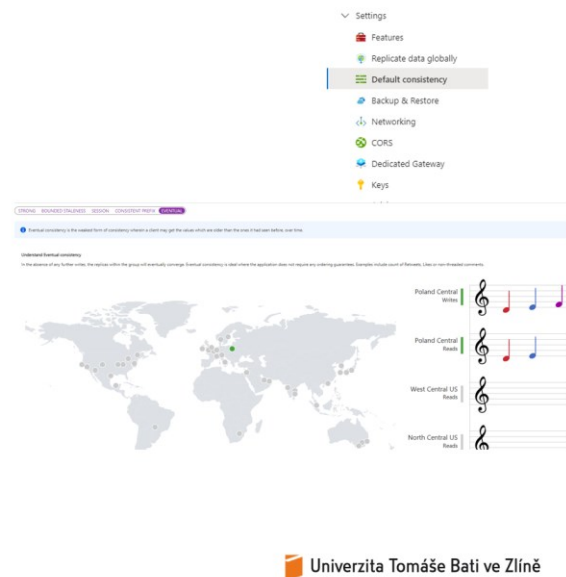
## Konfigurace škálování

Mapa znázorňuje globální distribuci datových center, kde jsou umístěny repliky dat.

Příklad ukazuje:

- Poland Central (zápisy): Zápisy dat jsou prováděny v datovém centru v Polsku.
- Poland Central (čtení): Čtení dat jsou prováděna v datovém centru v Polsku.
- West Central US (čtení): Čtení dat jsou prováděna v datovém centru v západní centrální části USA.
- North Central US (čtení): Čtení dat jsou prováděna v datovém centru v severní centrální části USA.

Hudební notace na pravé straně obrázku symbolizuje konzistenci dat a jak se data mohou lišit v různých regionech při použití eventuální konzistence.



Obrázek 14. Příklad slajdu z prezentaci. Konfigurace škálování.

### 3.4.5 5. týden: Instalace a nastavení nástrojů pro práci s Azure Cosmos DB

Prezentace "Instalace a nastavení nástrojů pro práci s Azure Cosmos DB" je zaměřena na poskytnutí podrobných instrukcí k instalaci a nastavení základních nástrojů potřebných pro práci s Azure Cosmos DB. Na začátku prezentace se diskutuje význam správných nástrojů pro efektivní práci s databázemi a aplikacemi. Poté následují konkrétní kroky k instalaci Visual Studio Code, což je lehký a výkonný editor kódu, podporující různé programovací jazyky. Následuje instalace Node.js, runtime prostředí pro spouštění JavaScript kódu mimo prohlížeč, včetně ověření instalace pomocí příkazů v terminálu.

Prezentace dále pokračuje kroky pro instalaci Azure Cosmos DB SDK, která usnadňuje práci s Cosmos DB přímo z kódu. Jsou zde popsány kroky pro inicializaci nového projektu Node.js a instalaci potřebného balíčku pomocí npm. V závěru prezentace je detailní návod pro připojení k Azure Cosmos DB z VS Code, včetně instalace Azure rozšíření, přihlášení do Azure účtu a připojení k Cosmos DB účtu přímo z editoru.

## Instalace Node.js



Co je Node.js?

- Node.js je JavaScript runtime prostředí, které umožňuje spouštět JavaScript kód mimo prohlížeč.

Kroky pro instalaci:

- 1) Navštivte oficiální stránku Node.js: [nodejs.org](https://nodejs.org)
- 2) Klikněte na "Download" a vyberte LTS verzi (Long Term Support) pro váš operační systém.
- 3) Spusťte stažený instalátor a postupujte podle pokynů instalačního průvodce.
- 4) Po instalaci ověřte instalaci v terminálu nebo příkazovém řádku pomocí příkazů:

```
node -v
```

```
npm -v
```

Ověření instalace:

- Příkazy by měly vrátit verzi nainstalovaného Node.js a NPM.

Obrázek 15. Příklad slajdu z prezentaci. Instalace Node.js.

### 3.4.6 6. týden: Přidávání dat do Azure Cosmos DB (Create)

Tato prezentace se soustředí na přidávání dat do Azure Cosmos DB. Začíná úvodem, kde je vysvětleno, proč je vkládání dat důležité, a stanovuje cíle cvičení, tedy naučit se základní postupy pro vkládání dat do Azure Cosmos DB pomocí NoSQL API a JavaScript SDK.

Následuje část o vytvoření databáze a kontejneru, kde jsou uvedeny kroky pro vytvoření databáze a kontejneru v Azure Portálu. Studenti se naučí, jak vytvořit nový účet Cosmos DB, otevřít "Data Explorer", vytvořit databázi a kontejner a nastavit partition key.

Další část se věnuje přípravě projektu v Node.js. Studenti se naučí, jak inicializovat nový projekt Node.js, nainstalovat Azure Cosmos DB SDK a vytvořit základní strukturu projektu.

Poté následuje popis kódu pro připojení k databázi. Tento kód slouží k připojení k Azure Cosmos DB a k získání referencí na specifickou databázi a kontejner. Jsou zde popsány jednotlivé kroky, jako je importování 'CosmosClient', definice 'endpoint' a 'key', vytvoření instance 'CosmosClient' a získání referencí na konkrétní databázi a kontejner.

Prezentace pokračuje vysvětlením, kde najít 'endpoint' a 'key' pro připojení k Azure Cosmos DB, a jak vytvořit funkci pro vkládání dat. Studenti se naučí, jak přidat nové dokumenty do kontejneru pomocí funkce addItem.

Následuje část o vkládání dat do kontejneru, kde je definováno pole items obsahující data, která mají být vložena do databáze. Jsou zde uvedeny jednotlivé kroky pro vložení dat a kontrolu správného vložení v Azure Portálu.

Prezentace se uzavírá sekcí o provádění změn a kontrole, kde studenti ověří, že data byla správně uložena do databáze, a děkují za pozornost.

## Připojení k Azure Cosmos DB

Vytvoření souboru  
index.js:

- 1) Otevřete Visual Studio Code.
- 2) Vytvořte nový soubor s názvem index.js ve vašem projektu.
- 3) Vložte následující kód pro připojení k Azure Cosmos DB:

```
1 // Importujeme CosmosClient z balíčku @azure/cosmos
2 const { CosmosClient } = require("@azure/cosmos");
3
4 // Definujeme endpoint a key pro připojení k Azure Cosmos DB
5 const endpoint = "vsi_cosmos_db_endpoint"; // Nahraďte vaším skutečným endpointem
6 const key = "vsi_cosmos_db_klic"; // Nahraďte vaším skutečným klíčem
7
8 // Vytvoříme instanci CosmosClient s použitím endpointu a klíče
9 const client = new CosmosClient({ endpoint, key });
10
11 // Definujeme ID databáze a kontejneru
12 const databaseId = "StudentDatabase";
13 const containerId = "StudentContainer";
14
15 // Získáme reference na databázi a kontejner
16 const database = client.database(databaseId);
17 const container = database.container(containerId);
18
19 // Hlavní funkce, kde budeme provádět naše operace
20 async function main() {
21   // Vaše operace zde
22 }
23
24 // Spuštění hlavní funkce a ošetření případných chyb
25 main().catch(err => {
26   console.error(err);
27 });
28
```

 Univerzita Tomáše Bati ve Zlíně

Obrázek 16. Příklad slajdu z prezentaci. Připojení k Azure Cosmos DB.

### 3.4.7 7. týden: Aktualizace dat v Azure Cosmos DB (Update)

Tato prezentace je věnována důležitosti aktualizace dat a naučí studenty, jak provádět aktualizace v Azure Cosmos DB pomocí NoSQL API a JavaScript SDK. Aktualizace dat umožňuje udržovat informace aktuální a přesné, opravovat chyby a přizpůsobovat data novým požadavkům.

Nejdříve se studenti seznámí s funkcí updateItem, která slouží k aktualizaci existujícího dokumentu v kontejneru v Azure Cosmos DB. Funkce přijímá dva parametry: id (unikátní identifikátor dokumentu) a updates (objekt obsahující nové hodnoty). Proces aktualizace zahrnuje načtení existujícího dokumentu, aktualizaci s novými hodnotami a uložení zpět do kontejneru.

Poté je prezentován kód, který definuje objekt updates obsahující data, která mají být aktualizována v existujících dokumentech. Každý prvek v poli updates obsahuje id

dokumentu a nové hodnoty atributů. Například dokument s id "1" bude aktualizován změnou age na 26 a city na Brno.

Prezentace dále vysvětluje, jak provést změny ve Visual Studio Code. Studenti vytvoří nový soubor index.js, vloží kód pro připojení k Azure Cosmos DB a spustí skript pomocí příkazu node index.js v terminálu. Poté ověří, že se data správně vložila do databáze prostřednictvím Azure Portálu a "Data Explorer".

Konečně, studenti ověří, že data byla úspěšně změněna na stránce Cosmos DB, což je klíčové pro pochopení procesu aktualizace a jeho významu v reálném prostředí.

Tato cvičení poskytují praktické dovednosti potřebné pro efektivní správu a aktualizaci dat v Azure Cosmos DB, což je důležité pro udržování aktuálnosti a přesnosti informací v databázových systémech.

## Funkce pro aktualizaci dat

Tento kód definuje funkci updateItem, která slouží k aktualizaci existujícího dokumentu v kontejneru v Azure Cosmos DB. Funkce přijímá dva parametry: id (unikátní identifikátor dokumentu, který chceme aktualizovat) a updates (objekt obsahující nové hodnoty, které chceme do dokumentu vložit). Funkce postupuje následujícím způsobem:

- Nejprve definujeme 'partitionKey' jako 'id', protože v našem případě je identifikátor dokumentu použit jako 'PartitionKey'.
- Pomocí 'container.item(id, partitionKey).read()' načteme existující dokument z kontejneru. Výsledek této operace je uložen do proměnné 'item'. Pokud dokument neexistuje, vypíšeme zprávu do konzole a funkce se ukončí.
- Pokud dokument existuje, použijeme 'Object.assign(item, updates)' k aktualizaci dokumentu novými hodnotami z objektu 'updates'.
- Pomocí 'container.item(id, partitionKey).replace(item)' uložíme aktualizovaný dokument zpět do kontejneru. Výsledek této operace je uložen do proměnné 'updatedItem'.
- Nakonec vypíšeme do konzole zprávu potvrzující úspěšnou aktualizaci dokumentu.
- Pokud během aktualizace dojde k chybě, vypíšeme do konzole chybovou zprávu obsahující podrobnosti o chybě.

Tato funkce je užitečná pro provádění změn v již existujících dokumentech v Azure Cosmos DB, aniž by bylo nutné dokumenty nejprve ručně mazat a znovu vytvářet.

```
// Funkce pro aktualizaci existujícího dokumentu v kontejneru
async function updateItem(id, updates) {
  try {
    // Použijeme studentID jako PartitionKey
    const partitionKey = id;
    // Načteme existující dokument
    const { resource: item } = await container.item(id, partitionKey).read();
    if (!item) {
      console.log('Item with id ${id} not found');
      return;
    }
    // Aktualizujeme dokument s novými hodnotami
    Object.assign(item, updates);
    // Uložíme aktualizovaný dokument zpět do kontejneru
    const { resource: updatedItem } = await container.item(id, partitionKey).replace(item);
    console.log('Updated item with id: ${updatedItem.id}');
  } catch (error) {
    console.error('Failed to update item with id ${id}: ${error.message}');
  }
}
```

Obrázek 17. Příklad slajdu z prezentaci. Funkce pro aktualizace dat.

### 3.4.8 8. týden: Mazání dat v Azure Cosmos DB (Delete)

8. cvičení kurzu Azure Cosmos DB se zaměřuje na mazání dat pomocí API a JavaScript SDK. V úvodu se studenti dozvědí, proč je mazání dat důležité pro udržování databáze aktuální a efektivní. Poté se naučí základní postupy pro mazání dat v Azure Cosmos DB. Cvičení obsahuje funkci pro mazání dat, která kontroluje existenci dokumentu před jeho smazáním a zajišťuje správné odstranění z kontejneru Cosmos DB. Studenti budou pracovat



s kódem, který definuje identifikátory dokumentů k odstranění, a ověří úspěšné smazání dat v Azure Portálu. Na závěr se provede kontrola, zda byla data úspěšně smazána.

## Funkce pro mazání dat

Tento kód definuje asynchronní funkci `deleteItem`, která je určena k odstranění existujícího dokumentu v kontejneru Azure Cosmos DB. Zde je podrobnější popis jednotlivých částí funkce:

- Funkce přijímá parametr `'id'`, což je identifikátor dokumentu, který chceme smazat.
  - Definuje proměnnou `'partitionKey'`, kterou nastavuje na hodnotu `'id'`, protože v tomto případě používáme `'studentId'` jako partition key.
  - Pokusí se načíst dokument s daným `'id'` a `'partitionKey'` pomocí metody `'container.item(id, partitionKey).read()'`.
  - Pokud dokument neexistuje (item je undefined), vypíše zprávu do konzole a ukončí funkci.
  - Pokud dokument existuje, zavolá metodu `'delete()'` na objektu `'container.item(id, partitionKey)'`, což vede k odstranění dokumentu.
  - Po úspěšném odstranění dokumentu vypíše zprávu do konzole s potvrzením.
  - V případě chyby (například pokud dokument nelze načíst nebo smazat), zachytí chybu a vypíše chybovou zprávu do konzole.
- Tato funkce tedy efektivně kontroluje existenci dokumentu před jeho smazáním a zajišťuje, že dokument je správně odstraněn z kontejneru Cosmos DB.

```
// Funkce pro smazání existujícího dokumentu v kontejneru
async function deleteItem(id) {
  try {
    // Použijeme studentId jako PartitionKey
    const partitionKey = id;
    // Smazeme dokument s daným id a partitionKey
    await container.item(id, partitionKey).delete();
    console.log('Deleted item with id: ' + id);
  } catch (error) {
    console.error('Failed to delete item with id ' + id + ': ' + error.message);
  }
}
```

 Univerzita Tomáše Bati ve Zlíně

Obrázek 18. Příklad slajdu z prezentaci. Funkce pro mazání dat.

### 3.4.9 9. týden: Čtení dat a práce s daty v Azure Cosmos DB pomocí JavaScriptu (Read)

Prezentace zaměřená na čtení dat a práci s daty v Azure Cosmos DB pomocí JavaScriptu. Úvodní část se věnuje důležitosti čtení dat, které umožňuje získávání uložených informací z databáze a jejich analýzu a zobrazení. Studenti se naučí základní postupy pro čtení dat v Azure Cosmos DB pomocí NoSQL API a JavaScript SDK a prozkoumají různé možnosti práce s daty. Prezentace obsahuje kód definující asynchronní funkci `'queryItems'`, která provádí SQL dotazy na kontejner Azure Cosmos DB a vrací výsledky. Funkce `'queryItems'` je podrobně vysvětlena včetně jednotlivých částí kódu. Další část prezentace ukazuje, jak číst konkrétní dokument podle ID a jak provádět SQL dotazy pro čtení dat z Azure Cosmos DB. Studenti se naučí provádět změny a ověřovat, že se data správně vložila do databáze prostřednictvím Azure Portálu. Prezentace zahrnuje také praktické pokyny pro spuštění skriptu v terminálu a ověření výsledků v Azure Portálu.

## Příklad čtení dat v Cosmos DB

Tento kód ukazuje, jak číst konkrétní dokument podle ID a jak provádět SQL dotazy pro čtení dat z Azure Cosmos DB.

- 'const item = await readItem("1");': Tento řádek volá asynchronní funkci 'readItem' s parametrem "1", což je ID dokumentu, který chceme přečíst. Funkce 'readItem' načte dokument s daným ID z Cosmos DB.
- 'console.log(item);': Tento řádek vypíše načtený dokument do konzole.
- 'const query = "SELECT \* from c WHERE c.age >= 25";': Tento řádek definuje SQL dotaz, který vybírá všechny dokumenty z kontejneru, kde hodnota pole 'age' je větší nebo rovna 25.
- 'const items = await queryItems(query);': Tento řádek volá asynchronní funkci 'queryItems' s parametrem 'query', což je SQL dotaz definovaný v předchozím kroku. Funkce 'queryItems' provede dotaz a vrátí výsledky.
- 'console.log(items);': Tento řádek vypíše výsledky dotazu do konzole.

Kód ukazuje základní operace pro čtení konkrétního dokumentu podle ID a provádění SQL dotazu pro čtení dat z Azure Cosmos DB.

```
// Hlavní funkce, kde budeme provádět čtení
async function main() {
  // Čtení konkrétního dokumentu podle ID
  const item = await readItem("1");
  console.log(item);

  // Čtení dat pomocí SQL dotazu
  const query = "SELECT * from c WHERE c.age >= 25";
  const items = await queryItems(query);
  console.log(items);
}
```

 Univerzita Tomáše Bati ve Zlíně

Obrázek 19. Příklad slajdu z prezentaci. Příklad čtení dat v Cosmos DB.

### 3.4.10 10. týden: Obhajoba projektů

10. cvičení kurzu Azure Cosmos DB je zaměřeno na obhajobu projektů, které studenti vypracovali během kurzu. Úvodní část prezentace vysvětluje důležitost obhajoby projektů, která umožňuje studentům prezentovat své znalosti a dovednosti, získat zpětnou vazbu od instruktorů a spolužáků a posílit jejich schopnosti v komunikaci a prezentaci technických projektů. Cílem cvičení je připravit studenty na obhajobu jejich projektů, vysvětlit postup a kritéria hodnocení. Prezentace se zaměřuje na to, co by měla obhajoba projektu obsahovat, včetně úvodu a cíle projektu, použitých technologií a nástrojů, implementace hlavních funkcí a struktury kódu, prezentace výsledků a praktických ukázek a shrnutí zkušeností získaných během práce na projektu. Dále prezentace vysvětluje postup obhajoby, která bude zahrnovat prezentaci projektu (5 minut), demonstraci (2 minuty) a otázky a odpovědi (5 minut). Kritéria hodnocení projektů zahrnují technickou úroveň, kvalitu kódu, použití správných technologií a nástrojů, funkčnost a uživatelské rozhraní, originalitu a inovaci a jasnost a srozumitelnost prezentace. Prezentace končí poděkováním za účast v seminářích a za projekty studentů.

## Kritéria hodnocení

Kritéria hodnocení projektů:

Technická úroveň

- Kvalita kódu, použití správných technologií a nástrojů.

Funkčnost a uživatelské rozhraní

- Jak dobře projekt funguje a jaké má uživatelské rozhraní.

Originalita a inovace

- Kreativita a inovativní přístup k řešení problémů.

Prezentace

- Jasnost a srozumitelnost prezentace.

Obrázek 20. Příklad slajdu z prezentaci. Kritéria hodnocení zápočtového projektu.

### 3.5 Absolvování předmětu

#### 3.5.1 Klasifikovaný zápočet

Závěrečný klasifikovaný zápočet je navržen tak, aby studenti mohli demonstrovat své praktické dovednosti a teoretické znalosti získané během kurzu o Azure Cosmos DB.

Tento úkol bude sloužit jako klíčový prvek pro posouzení jejich schopností a připravenosti využívat Cosmos DB v reálných aplikacích. Úspěšné dokončení zápočtu je nezbytné pro úspěšné zakončení předmětu a získání kreditů.

##### 3.5.1.1 Cíle zápočtu

- Aplikace teoretických znalostí

Ověřit, zda studenti rozumí klíčovým konceptům Azure Cosmos DB, jako je vytváření databází, konfigurace a správa dat.

- Praktické dovednosti

Zajistit, že studenti jsou schopni prakticky aplikovat CRUD operace a další databázové techniky.

##### 3.5.1.2 Požadavky pro klasifikovaný zápočet

- Vytvoření databáze a kontejneru

Studenti musí navrhnout a vytvořit databázi v Azure Cosmos DB, která bude odpovídat konkrétnímu účelu, jako je správa knihovny nebo evidence zákazníků.

Databáze musí být správně nakonfigurována a optimalizována pro očekávané zatížení.

- Implementace CRUD operací

Studenti musí implementovat základní CRUD operace: Vytvoření (Create), Čtení (Read), Aktualizace (Update) a Mazání (Delete) dat.

CRUD operace musí být efektivní a správně implementované podle zadaných požadavků.

- Dokumentace

Studenti musí předložit dokumentaci, která popisuje návrh databáze, použité technologie a implementovaná bezpečnostní opatření.

Dokumentace by měla obsahovat příklady kódu a vysvětlení jednotlivých funkcí a nastavení.

- Presentace projektu

Studenti musí prezentovat svůj projekt před vyučujícím a případně před spolužáky.

Prezentace by měla zahrnovat ukázkou funkčnosti databáze, vysvětlení klíčových rozhodnutí a implementovaných funkcí.

### **3.5.1.3 Hodnocení**

Projekt bude hodnocen podle následujících kritérií:

- Funkčnost a kompletnost: Zda projekt splňuje všechny požadavky a zda všechny CRUD operace fungují správně.
- Kvalita kódu: Efektivita, čitelnost a organizace kódu.
- Bezpečnostní opatření: Úroveň a implementace bezpečnostních opatření.
- Kvalita dokumentace a prezentace: Srozumitelnost, úplnost a profesionalita dokumentace a prezentace.

### **3.5.1.4 Závěr**

Tento klasifikovaný zápočet je navržen tak, aby komplexně prověřil schopnosti studentů v oblasti práce s Azure Cosmos DB. Úspěšné dokončení projektu a získání zápočtu prokazuje, že studenti jsou připraveni aplikovat své znalosti a dovednosti v reálných projektech a profesních situacích, což je klíčové pro jejich další rozvoj v oblasti informačních technologií.

### 3.5.2 Zadání pro studenty: Vytvoření malé databáze s CRUD operacemi v Azure Cosmos DB

V tomto zadání budete vytvářet jednoduchou databázi v Azure Cosmos DB, která bude sloužit k ukládání informací o knihách v knihovně. Vaším úkolem bude vytvořit databázi, kontejner a implementovat základní CRUD operace (Create, Read, Update, Delete) pro správu těchto dat.

#### Cíle

- Vytvoření Azure Cosmos DB účtu a databáze.
- Vytvoření kontejneru pro ukládání informací o knihách.
- Implementace základních CRUD operací pomocí NoSQL API a JavaScriptu.

#### Popis úkolu

1. Vytvoření Azure Cosmos DB účtu
  - Vytvořte nový účet Azure Cosmos DB.
2. Vytvoření databáze a kontejneru
  - Vytvořte databázi.
  - Vytvořte kontejner s oddílem `/id`.
3. Implementace CRUD operací
  - Create (Vložení nového dokumentu): Vložte nový dokument s informacemi.
  - Read (Čtení dokumentů): Načtěte a zobrazte informace z databáze.
  - Update (Aktualizace dokumentu): Aktualizujte existující dokument s novými informacemi.
  - Delete (Smazání dokumentu): Odstraňte dokument z databáze.
4. Testování a ověření
  - Ujistěte se, že CRUD operace fungují správně tím, že vložíte několik dat do databáze, provedete dotazy pro čtení dat, aktualizujete některé záznamy a odstraníte záznamy podle potřeby.

#### Formát a požadavky na odevzdání projektu

1. Zdrojový kód projektu
  - Všechny zdrojový kód by měl být nahrán do jedné komprimované složky (ZIP nebo TAR).

- Složka by měla obsahovat veškerý kód potřebný pro vytvoření a správu databáze, včetně skriptů pro CRUD operace a jakékoliv další pomocné skripty.
  - Struktura složek by měla být přehledná a dobře organizovaná.
2. Dokumentace projektu
- Dokumentace by měla být ve formátu PDF.
  - Dokumentace by měla obsahovat:
    - a) Úvod a přehled projektu.
    - b) Popis architektury databáze.
    - c) Podrobný popis implementovaných funkcí a kódu.
    - d) Příklady použití CRUD operací.
    - e) Bezpečnostní opatření a konfigurace.
    - f) Závěry a případné návrhy na zlepšení.
  - Dokumentace by měla být jasná, strukturovaná a obsahovat všechny požadované části.
3. Prezentace projektu
- Prezentace by měla být ve formátu PDF nebo PPTX.
  - Měla by obsahovat klíčové body projektu, jako je architektura databáze, hlavní funkce, ukázky kódu a výsledky.
  - Prezentace by měla být stručná, přehledná a vizuálně přitažlivá.

## 4 CASE STUDY: NEWDAY A IMPLEMENTACE AZURE COSMOS DB

### 4.1 Úvod

NewDay, jeden z předních poskytovatelů spotřebitelského úvěru ve Velké Británii, je známý svým inovativním přístupem k poskytování nezajištěných úvěrů široké škále spotřebitelů. Společnost, která slouží téměř 5 milionům zákazníků, spolupracuje s významnými spotřebitelskými značkami jako John Lewis & Partners, Argos a Deko a vydává jednu z pěti kreditních karet ve Velké Británii. V roce 2022 se NewDay podařilo zlepšit kreditní skóre přibližně 2 milionům zákazníků. NewDay se zaměřuje na technologicky pokročilé, flexibilní a inovativní produkty, což z nich dělá jednoho z největších poskytovatelů spotřebitelského úvěru v zemi. [17]

### 4.2 Výzvy a cíle

Historicky byla infrastruktura NewDay omezena na jednu platformu, což komplikovalo škálování a mezinárodní expanzi. Původní cíl společnosti byl přesvědčit jednotlivé maloobchodní značky k nákupu úvěrových produktů. Jak společnost rostla, náklady, výkon a škálovatelnost platformy se staly vážnými obavami.

NewDay si uvědomila, že finanční trh potřebuje technologii, která by umožnila značkám přímo komunikovat se zákazníky a poskytovat úvěry. Cílem bylo najít řešení, které by umožňovalo okamžitou škálovatelnost, nízkou latenci, globální distribuci a nejvyšší možnou dostupnost při zajištění výkonu na jakékoli úrovni. [17]

### 4.3 Řešení: Integrace Azure Cosmos DB

NewDay se rozhodla pro integraci Azure Cosmos DB, která je globálně distribuovanou databází, umožňující rychlé škálování a poskytující nízkou latenci při vysoké dostupnosti. Azure Cosmos DB byla klíčová pro experimentování s novými typy dat a různými doménami, čímž podporovala inovace ve společnosti. Díky Azure Cosmos DB mohl NewDay snadno přidávat nové typy dat a domény.

Společnost NewDay použila Azure Databricks pro hladkou migraci dat s nulovým výpadkem. To zahrnovalo přepracování kontejnerů Cosmos DB a dosažení nulového

výpadku během migrace. Azure Databricks umožnil bezpečnou a efektivní migraci dat, zatímco platforma zůstala v plné produkční zátěži. [17]

#### 4.4 Implementace a technické úpravy

Během procesu implementace bylo zapotřebí přepracovat kontejnery Azure Cosmos DB, aby bylo dosaženo nulového výpadku, a použít Azure Databricks k bezproblémové migraci kontejnerů. NewDay byla schopna upravit databázi tak, aby odpovídala různým pracovním zatížením a alokovala zdroje tam, kde byly nejvíce potřebné. Společnost také využila možnosti autoskalování Azure Cosmos DB pro dosažení nákladové efektivity. [17]

#### 4.5 Výsledky a dopady

Díky Azure Cosmos DB a přidruženým technologiím NewDay nejenže zvýšila efektivitu a spolehlivost svých operací, ale také se transformovala na dvě společnosti: technologickou firmu poskytující platformu jiným finančním institucím a NewDay Cards. Výsledná platforma nabízí embedded financování a end-to-end technologická řešení pro klienty jako Lloyds Banking Group a Boohoo. [17]

NewDay nyní poskytuje end-to-end řešení pro finanční instituce, což zahrnuje technologie, underwriting a další služby. Platforma podporuje partnerské spolupráce, včetně embedded finančních služeb a digitálních finančních produktů. [17]

#### 4.6 Závěr

NewDay úspěšně demonstrovala, jak může integrace moderních cloudových technologií jako Azure Cosmos DB radikálně transformovat a zlepšit finanční služby. Inovativní využití dat a technologií umožnilo společnosti rozšířit své operace a nabídnout lepší služby svým zákazníkům. NewDay tak nejenže zvýšila svou vlastní efektivitu, ale také poskytla své technologické řešení dalším finančním institucím, čímž posílila svou pozici na trhu. [17]



## 5 TESTOVACÍ ÚKOLY PRO AZURE COSMOS DB - MOODLE FORMÁT

### 5.1 Úkol 1: Vložení dokumentu

**Otázka:** Jak vložíte nový dokument do kontejneru pomocí Node.js SDK? Uveďte příklad kódu.

**Odpověď:**

```
const { CosmosClient } = require("@azure/cosmos");

const endpoint = "váš_cosmos_db_endpoint";

const key = "váš_cosmos_db_klíč";

const client = new CosmosClient({ endpoint, key });

const databaseId = "StudentDatabase";

const containerId = "StudentContainer";

const database = client.database(databaseId);

const container = database.container(containerId);

async function addItem(item) {

    item.studentId = item.id;

    const { resource: createdItem } = await container.items.create(item);

    console.log(`Created item with id: ${createdItem.id}`);

}

const item = { id: "1", name: "Jan Novak", age: 25, city: "Praha" };

addItem(item);
```

### 5.2 Úkol 2: Aktualizace dokumentu

**Otázka:** Popište, jak aktualizovat existující dokument v Azure Cosmos DB pomocí Node.js SDK. Uveďte příklad kódu.

**Odpověď:**

```
async function updateItem(id, updates) {
```

```
const partitionKey = id;

const { resource: item } = await container.item(id, partitionKey).read();

if (!item) {

  console.log(`Item with id ${id} not found`);

  return;

}

Object.assign(item, updates);

const { resource: updatedItem } = await container.item(id, partitionKey).replace(item);

console.log(`Updated item with id: ${updatedItem.id}`);

}

const updates = { id: "1", age: 26, city: "Brno" };

updateItem("1", updates);
```

### 5.3 Úkol 3: Smazání dokumentu

**Otázka:** Jak odstraníte dokument z Azure Cosmos DB pomocí Node.js SDK? Uveďte příklad kódu.

**Odpověď:**

```
async function deleteItem(id) {

  const partitionKey = id;

  const { resource: item } = await container.item(id, partitionKey).read();

  if (!item) {

    console.log(`Item with id ${id} not found`);

    return;

  }

  await container.item(id, partitionKey).delete();

  console.log(`Deleted item with id: ${id}`);

}
```

```
deleteItem("1");
```

## 5.4 Úkol 4: Čtení dat pomocí SQL dotazů

**Otázka:** Napište SQL dotaz pro získání všech dokumentů, kde je věk větší než 25, a implementujte jej pomocí Node.js SDK.

**Odpověď:**

```
async function queryItems(query) {  
    const { resources: items } = await container.items.query(query).fetchAll();  
    console.log(`Query returned ${items.length} items`);  
    return items;  
}  
  
const query = "SELECT * from c WHERE c.age > 25";  
  
const items = await queryItems(query);  
  
console.log(items);
```

## 5.5 Úkol 5: Migrace dat do Azure Cosmos DB

**Otázka:** Jaké jsou hlavní kroky při migraci dat do Azure Cosmos DB? Uveďte alespoň 3 kroky.

**Odpověď:**

**Příprava dat:** Zkontrolujte a upravte strukturu vašich dat, aby byla kompatibilní s Azure Cosmos DB.

**Výběr nástroje pro migraci:** Vyberte vhodný nástroj pro migraci, jako je Azure Data Factory nebo Cosmos DB Data Migration Tool.

**Konfigurace a spuštění migrace:** Nastavte migrační proces v zvoleném nástroji a spusťte migraci. Sledujte průběh a ověřte správnost migrovaných dat.

## 6 TESTOVACÍ OTÁZKY Z TEORIE AZURE COSMOS DB

### 6.1 1. týden: Úvod do Azure Cosmos DB

Co je Azure Cosmos DB?

- a) Textový editor
- b) Multimodelová databázová služba
- c) Operační systém
- d) Síťový protokol

**Správná odpověď: b) Multimodelová databázová služba**

Které z následujících funkcí patří mezi klíčové vlastnosti Azure Cosmos DB?

- a) Vysoká dostupnost
- b) Globální distribuce dat
- c) Podpora různých datových modelů
- d) Vše výše uvedené

**Správná odpověď: d) Vše výše uvedené**

### 6.2 2. týden: Datové modely a API

Který z následujících datových modelů není podporován Azure Cosmos DB?

- a) SQL API
- b) Gremlin API
- c) Table API
- d) XML API

**Správná odpověď: d) XML API**

Jaký je typický scénář použití pro Gremlin API?

- a) Správa grafových databází
- b) Dotazování v relačních databázích
- c) Ukládání klíč-hodnota párů

d) Sloupcově orientované databáze

**Správná odpověď: a) Správa grafových databází**

### **6.3 3. týden: Konzistence dat**

**Která z následujících úrovní konzistence zajišťuje nejvyšší úroveň konzistence dat v Azure Cosmos DB?**

a) Eventuální konzistence

b) Relativní konzistence

c) Silná konzistence

d) Session konzistence

**Správná odpověď: c) Silná konzistence**

**Jaká úroveň konzistence v Azure Cosmos DB poskytuje nejvyšší dostupnost a výkon?**

a) Silná konzistence

b) Eventuální konzistence

c) Relativní konzistence

d) Omezená stálá konzistence

**Správná odpověď: b) Eventuální konzistence**

### **6.4 4. týden: Škálování a dostupnost**

**Co je horizontální škálování v kontextu Azure Cosmos DB?**

a) Zvyšování výkonu přidáním více procesorů do jednoho serveru

b) Zvyšování výkonu přidáním dalších serverů

c) Snížení výkonu odstraněním nepotřebných serverů

d) Zlepšení výkonu zvýšením frekvence procesoru

**Správná odpověď: b) Zvyšování výkonu přidáním dalších serverů**

**Co zajišťuje vysokou dostupnost Azure Cosmos DB?**

a) Použití jednoho serveru

b) Globální distribuce dat a replikace

- c) Použití pouze lokálních disků
- d) Omezená podpora datových modelů

**Správná odpověď: b) Globální distribuce dat a replikace**

## **6.5 5. týden: Indexace a dotazování**

**Jaký je hlavní účel indexace v Azure Cosmos DB?**

- a) Zvýšení množství uložených dat
- b) Zrychlení dotazování a zlepšení výkonu
- c) Zvýšení počtu uživatelů
- d) Snížení množství potřebného úložného prostoru

**Správná odpověď: b) Zrychlení dotazování a zlepšení výkonu**

**Který dotazovací jazyk se používá pro dotazování v dokumentově orientovaném modelu (SQL API) v Azure Cosmos DB?**

- a) SPARQL
- b) T-SQL
- c) Gremlin
- d) SQL

**Správná odpověď: d) SQL**

## **6.6 6. týden: Bezpečnost a správa přístupu**

**Co znamená šifrování dat v kontextu Azure Cosmos DB?**

- a) Změna struktury dat
- b) Ochrana dat před neoprávněným přístupem
- c) Zrychlení přenosu dat
- d) Komprimace dat pro úsporu místa

**Správná odpověď: b) Ochrana dat před neoprávněným přístupem**

**Jaký mechanismus používá Azure Cosmos DB pro správu uživatelů a jejich přístupových práv?**

- a) Role-Based Access Control (RBAC)
- b) Access Control Lists (ACL)
- c) Simple Authentication and Security Layer (SASL)
- d) Password Authentication Protocol (PAP)

**Správná odpověď: a) Role-Based Access Control (RBAC)**

## **6.7 7. týden: Monitorování a ladění výkonu**

**Který nástroj v Azure je primárně používán pro monitorování výkonu Azure Cosmos DB?**

- a) Azure Monitor
- b) Azure DevOps
- c) Azure Active Directory
- d) Azure Backup

**Správná odpověď: a) Azure Monitor**

**Které metriky jsou klíčové pro sledování výkonu Azure Cosmos DB?**

- a) RU/s, latence, propustnost
- b) Počet uživatelů, velikost databáze, rychlost disku
- c) Počet serverů, využití paměti, teplota CPU
- d) Šířka pásma, doba odezvy, počet požadavků za sekundu

**Správná odpověď: a) RU/s, latence, propustnost**

## **6.8 8. týden: Integrace s dalšími Azure službami**

**Co je Azure Functions v kontextu Azure Cosmos DB?**

- a) Nástroj pro vizualizaci dat
- b) Serverless výpočetní služba pro běh kódu
- c) Úložiště pro zálohy databází
- d) Nástroj pro správu uživatelů a rolí

**Správná odpověď: b) Serverless výpočetní služba pro běh kódu**

**Jaká je hlavní výhoda integrace Azure Cosmos DB s Azure Logic Apps?**

- a) Zrychlení dotazování
- b) Automatizace pracovních postupů
- c) Zvýšení bezpečnosti
- d) Snížení nákladů na úložiště

**Správná odpověď: b) Automatizace pracovních postupů**

## **6.9 9. týden: Migrace dat do Cosmos DB**

**Který nástroj lze použít pro migraci dat do Azure Cosmos DB?**

- a) Azure Data Factory
- b) Azure DevOps
- c) Azure Active Directory
- d) Azure Monitor

**Správná odpověď: a) Azure Data Factory**

**Co je klíčové při plánování migrace dat do Azure Cosmos DB?**

- a) Výběr správné strategie migrace
- b) Zvýšení počtu uživatelů
- c) Snížení velikosti databáze
- d) Změna struktury dat

**Správná odpověď: a) Výběr správné strategie migrace**

## **6.10 10. týden: Case studies a best practices**

**K čemu slouží case studies v kontextu Azure Cosmos DB?**

- a) Ukázky úspěšných implementací a získané poznatky
- b) Analýza bezpečnostních rizik
- c) Testování nových funkcí
- d) Školení nových uživatelů



**Správná odpověď: a) Ukázky úspěšných implementací a získané poznatky**

**Co je best practice pro optimalizaci výkonu v Azure Cosmos DB?**

- a) Používání pouze jednoho datového modelu
- b) Implementace správné indexace a optimalizace dotazů
- c) Snížení počtu uživatelů
- d) Zvýšení velikosti databáze

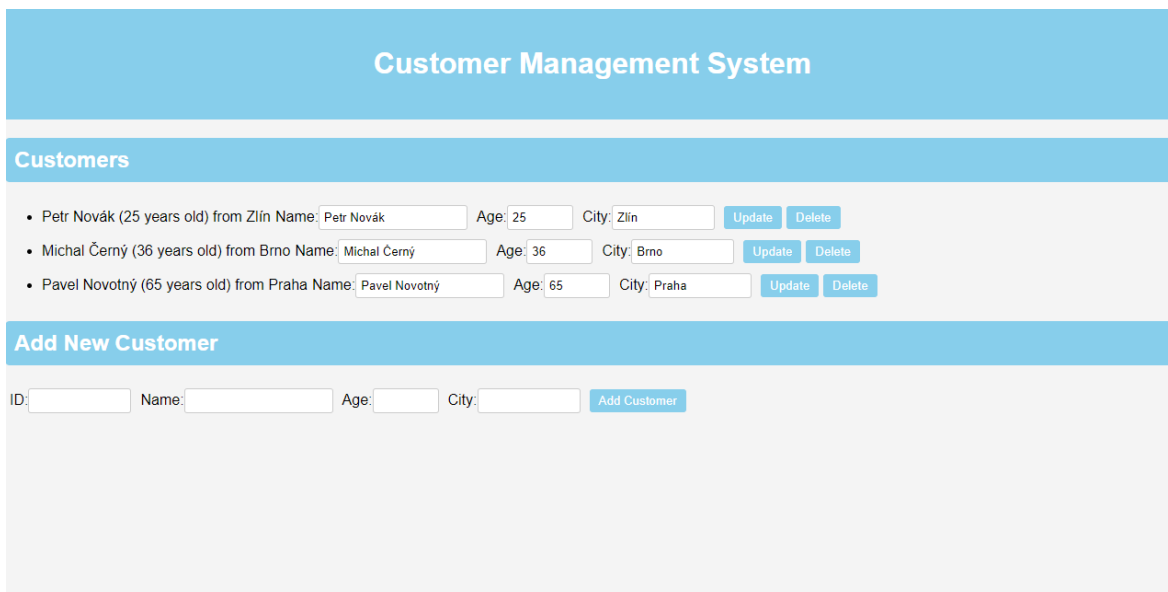
**Správná odpověď: b) Implementace správné indexace a optimalizace dotazů**

## 7 VZOROVÝ PŘÍKLAD + TUTORIÁL

Porozumění tomu, jak využívat teoretické znalosti v praxi, je klíčové. Aplikace teorie na reálné problémy zvyšuje praktickou hodnotu získaných vědomostí a rozvíjí schopnosti řešit problémy. Praktické dovednosti jsou nezbytné pro úspěšné uplatnění na pracovním trhu. Tento příklad s tutoriálem vám umožní využít znalosti práce s databázemi na konkrétním projektu, což povede k hlubšímu porozumění a zvýšení sebevědomí při práci s technologií.

### 7.1 Web pro správu databáze zákazníků

V tomto projektu bude vytvořena jednoduchá webová aplikace pro správu databáze zákazníků pomocí technologie Node.js, Express.js a šablonovacího jazyka Pug. Aplikace umožní provádět základní operace CRUD (Create, Read, Update, Delete) nad databází Cosmos DB. Tento web bude navržen tak, aby byl jednoduchý a přehledný, s intuitivním uživatelským rozhraním.



The screenshot shows a web application interface for a Customer Management System. At the top, there is a blue header with the text "Customer Management System". Below this, there is a section titled "Customers" which displays a list of three customer records. Each record includes the customer's name, age, and city, along with "Update" and "Delete" buttons. The records are: Petr Novák (25 years old) from Zlín, Michal Černý (36 years old) from Brno, and Pavel Novotný (65 years old) from Praha. Below the list, there is a section titled "Add New Customer" which contains input fields for ID, Name, Age, and City, and an "Add Customer" button.

Obrázek 21. Příklad webu pro správu databáze zákazníků pomocí Cosmos DB se základními CRUD operacemi.

#### Hlavní funkce webu:

- Přidávání nových zákazníků (Create):

Uživatelé budou moci přidávat nové zákazníky do databáze prostřednictvím jednoduchého formuláře. Stačí vyplnit jméno, věk a město zákazníka a odeslat formulář. Nový záznam se okamžitě uloží do Cosmos DB a zobrazí se na hlavní stránce.

- Zobrazení všech zákazníků (Read):

Hlavní stránka aplikace bude zobrazovat seznam všech zákazníků uložených v databázi. Každý zákazník bude zobrazen s jeho jménem, věkem a městem.

- Aktualizace existujících zákazníků (Update):

Uživatelé budou moci aktualizovat údaje existujících zákazníků. Kliknutím na tlačítko "Edit" vedle záznamu zákazníka se otevře formulář s předvyplněnými údaji, které bude možné upravit a uložit zpět do databáze.

- Mazání zákazníků (Delete):

Uživatelé budou moci mazat záznamy zákazníků kliknutím na tlačítko "Delete" vedle příslušného záznamu. Záznam bude odstraněn z databáze a již se nezobrazí na hlavní stránce.

Tato aplikace bude sloužit jako základní nástroj pro správu zákazníků a demonstruje, jak efektivně používat technologii Cosmos DB v kombinaci s Node.js a Express.js. Následující část obsahuje podrobný tutoriál, který vás provede krok za krokem celým procesem tvorby této aplikace.

## 7.2 Tutoriál

### Krok 1: Instalace Node.js a vytvoření projektu

- 1) Nainstalujte Node.js a npm, pokud ještě nejsou nainstalovány.
- 2) Vytvořte novou složku pro projekt a přejděte do ní pomocí Terminálu Windows:

```
mkdir customer-management
```

```
cd customer-management
```

- 3) Inicializujte nový projekt npm:

```
npm init -y
```

### Krok 2: Instalace potřebných balíčků

- 1) Nainstalujte Express, Azure Cosmos DB SDK a další potřebné balíčky:

```
npm install express @azure/cosmos pug
```

### Krok 3: Vytvoření struktury složek

- 1) Vytvořte potřebné složky pro váš projekt:

```
mkdir views public
```

2) Vytvořte soubory pro váš projekt:

```
New-Item -Path index.js, public/style.css, views/index.pug, views/layout.pug -ItemType File
```

#### Krok 4: Nastavení Express

1) V 'index.js' přidejte kód pro nastavení Express a připojení k Azure Cosmos DB:

```
const express = require('express');
const { CosmosClient } = require('@azure/cosmos');
const app = express();
// Nastavení Pug jako šablonovacího systému
app.set('view engine', 'pug');
// Nastavení statických souborů
app.use(express.static('public'));
app.use(express.urlencoded({ extended: true }));
// Parametry připojení k Azure Cosmos DB
const endpoint = "https://your-cosmos-db-account.documents.azure.com:443/";
const key = "your-cosmos-db-key";
const client = new CosmosClient({ endpoint, key });
const databaseId = 'CustomerDatabase';
const containerId = 'CustomerContainer';
const database = client.database(databaseId);
const container = database.container(containerId);
// Hlavní stránka
app.get('/', async (req, res) => {
  const { resources: customers } = await container.items.query('SELECT * FROM c').fetchAll();
```

```
    res.render('index', { customers });
  });

// Přidání nového zákazníka
app.post('/add', async (req, res) => {
  const { id, name, age, city } = req.body;

  await container.items.create({ id, name, age: parseInt(age), city });

  res.redirect('/');
});

// Aktualizace informací o zákazníkovi
app.post('/update', async (req, res) => {
  const { id, name, age, city } = req.body;

  const partitionKey = id;

  const { resource: item } = await container.item(id, partitionKey).read();

  Object.assign(item, { name, age: parseInt(age), city });

  await container.item(id, partitionKey).replace(item);

  res.redirect('/');
});

// Smazání zákazníka
app.post('/delete', async (req, res) => {
  const { id } = req.body;

  const partitionKey = id;

  await container.item(id, partitionKey).delete();

  res.redirect('/');
});

// Spuštění serveru
const PORT = process.env.PORT || 3000;
```

```
app.listen(PORT, () => {  
  console.log(`Server is running on http://localhost:${PORT}`);  
});
```

### Krok 5: Vytvoření šablon

- 1) V 'views/layout.pug' přidejte základní rozvržení:

```
doctype html  
  
html  
  
  head  
  
    title Customer Management  
  
    link(rel='stylesheet', href='/style.css')  
  
  body  
  
    header  
  
      h1 Customer Management System  
  
    main  
  
      block content
```

- 2) V 'views/index.pug' přidejte šablonu pro zobrazení a správu zákazníků:

```
extends layout  
  
block content  
  
  h2(style='background-color: #87CEEB; padding: 10px; border-radius: 5px; color: white;')  
  Customers  
  
  ul  
  
    each customer in customers  
  
      li  
  
        | #{customer.name} (#{customer.age} years old) from #{customer.city}  
  
        form(action='/update', method='POST', style='display: inline-block; margin: 5px;')  
  
          input(type='hidden', name='id', value=customer.id)
```

label(for='name') Name:

```
input(type='text', name='name', value=customer.name, required, style='width: 150px; margin-right: 10px;')
```

label(for='age') Age:

```
input(type='number', name='age', value=customer.age, required, style='width: 60px; margin-right: 10px;')
```

label(for='city') City:

```
input(type='text', name='city', value=customer.city, required, style='width: 100px; margin-right: 10px;')
```

```
button(type='submit', style='background-color: #87CEEB; color: white; border: none; padding: 5px 10px; border-radius: 3px;') Update
```

```
form(action='/delete', method='POST', style='display:inline')
```

```
input(type='hidden', name='id', value=customer.id)
```

```
button(type='submit', style='background-color: #87CEEB; color: white; border: none; padding: 5px 10px; border-radius: 3px;') Delete
```

```
h2(style='background-color: #87CEEB; padding: 10px; border-radius: 5px; color: white;')  
Add New Customer
```

```
form(action='/add', method='POST', style='display: inline-block; margin: 5px;')
```

label(for='id') ID:

```
input(type='text', name='id', required, style='width: 100px; margin-right: 10px;')
```

label(for='name') Name:

```
input(type='text', name='name', required, style='width: 150px; margin-right: 10px;')
```

label(for='age') Age:

```
input(type='number', name='age', required, style='width: 60px; margin-right: 10px;')
```

label(for='city') City:

```
input(type='text', name='city', required, style='width: 100px; margin-right: 10px;')
```

```
button(type='submit', style='background-color: #87CEEB; color: white; border: none; padding: 5px 10px; border-radius: 3px;') Add Customer
```

**Krok 6: Nastavení CSS**

1) V public/style.css přidejte styly pro vylepšení vzhledu:

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
    margin: 0;  
    padding: 0;  
}  
  
header {  
    background-color: #87CEEB;  
    padding: 20px;  
    text-align: center;  
    color: white;  
}  
  
h2 {  
    background-color: #87CEEB;  
    padding: 10px;  
    border-radius: 5px;  
    color: white;  
}  
  
form {  
    margin: 10px 0;  
}  
  
input[type="text"],  
input[type="number"] {  
    padding: 5px;
```



```
margin-right: 10px;

border-radius: 3px;

border: 1px solid #ccc;
}

button {

background-color: #87CEEB;

color: white;

border: none;

padding: 5px 10px;

border-radius: 3px;

cursor: pointer;

}

button:hover {

background-color: #00BFFF;

}
```

### **Krok 7: Spuštění aplikace**

- 1) Spusťte aplikaci příkazem:

```
node index.js
```

### **Krok 8: Ověření funkčnosti**

Otevřete prohlížeč a přejděte na <http://localhost:3000>, abyste ověřili funkčnost vaší aplikace. Měli byste vidět stránku pro správu databáze zákazníků s možností přidávat, upravovat a mazat data.

## **7.3 Co je to Pug, Express a npm?**

Pug, Express a npm spolupracují na vytvoření moderních a dynamických webových aplikací. Pug poskytuje šablonovací jazyk, Express zajišťuje backendovou logiku a směrování, a npm spravuje závislosti a balíčky potřebné pro běh aplikace.

### 7.3.1 Pug

Pug je šablonovací jazyk, který umožňuje psát HTML stránky pomocí zjednodušené a zkrácené syntaxe. Pug byl původně znám jako Jade, ale kvůli problémům s ochrannou známkou musel být přejmenován. Pug se používá hlavně ve spojení s Node.js a Express.js pro tvorbu dynamických webových stránek.

#### Hlavní vlastnosti Pug:

- Zjednodušená syntaxe: Pug eliminuje potřebu psát uzavírající značky a udržuje kód čistý a přehledný.
- Indentace: Pug používá odsazení (indentaci) pro určení struktury dokumentu, což zjednodušuje čitelnost kódu.
- Dynamické vkládání dat: Pug umožňuje snadno vkládat dynamická data do šablon pomocí JavaScriptových proměnných a výrazů.
- Rozšířitelnost: Pug podporuje komponenty a mixiny, které umožňují opakované použití kódu a vytváření modulárních šablon.

### 7.3.2 Express

Express.js (nebo jednoduše Express) je minimalistický a flexibilní webový framework pro Node.js, který poskytuje sadu funkcí pro tvorbu webových a mobilních aplikací. Express zjednodušuje správu směrování (routing), middleware, a HTTP požadavky a odpovědi.

#### Hlavní vlastnosti Express:

- Rychlost a flexibilita: Express je navržen tak, aby byl rychlý a flexibilní, což umožňuje vývojářům snadno vytvářet aplikace různého rozsahu.
- Middleware: Express umožňuje použití middleware funkcí pro zpracování požadavků a odpovědí. Middleware jsou funkce, které mají přístup k požadavku (request) a odpovědi (response) objektům a mohou je upravovat nebo přidávat další funkce.
- Směrování (routing): Express poskytuje robustní systém směrování, který umožňuje mapovat URL adresy na konkrétní funkce nebo soubory.
- Šablonovací systémy: Express podporuje různé šablonovací systémy, jako je Pug, EJS, Handlebars a další, což umožňuje snadné vytváření dynamických HTML stránek.

### 7.3.3 npm

npm (Node Package Manager) je správce balíčků pro JavaScript a hlavní správce balíčků pro Node.js. Umožňuje vývojářům instalovat, sdílet a spravovat závislosti jejich projektů.

#### Hlavní vlastnosti npm:

- Instalace balíčků: npm umožňuje snadno instalovat knihovny a balíčky třetích stran, které lze použít ve vašem projektu. Instalace se provádí příkazem `npm install`.
- Správa závislostí: npm automaticky spravuje a sleduje závislosti vašeho projektu v souboru `package.json`. Tento soubor obsahuje seznam všech balíčků, které váš projekt používá, a jejich verze.
- Sdílení balíčků: Vývojáři mohou vytvářet vlastní balíčky a sdílet je s komunitou prostřednictvím npm registry. To umožňuje ostatním vývojářům snadno používat tyto balíčky ve svých projektech.
- Skriptování: npm umožňuje definovat a spouštět skripty přímo ze souboru `package.json`, což zjednodušuje běžné úkoly, jako je spuštění serveru, testování nebo build procesu.

## 8 MOŽNÉ SMĚRY ROZVOJE VÝUKOVÉHO PLÁNU

V této kapitole se zaměříme na možné směry rozvoje výukového plánu pro Azure Cosmos DB, aby studenti získali ještě hlubší a širší pochopení této technologie. Zde jsou návrhy na vylepšení a rozšíření stávajícího kurzu.

### 8.1 Pokročilé dotazování a indexace

Optimalizace dotazů: Studenti se mohou naučit pokročilé techniky pro optimalizaci dotazů a efektivnější využívání indexů.

Custom indexy: Jak vytvářet a spravovat vlastní indexy pro zlepšení výkonu dotazování. [4,18]

### 8.2 Práce s různými datovými modely

Hlubkové studium datových modelů: Detailní pochopení dokumentového, grafového, klíč-hodnota a sloupcového modelu, a kdy který model použít.

Pokročilé případy použití: Reálné scénáře a případové studie, které ukazují, jak různé datové modely řeší specifické problémy. [4,18]

### 8.3 Bezpečnost a správa přístupu

Detailní nastavení zabezpečení: Jak implementovat a spravovat šifrování dat, jaké jsou nejlepší praktiky pro zabezpečení dat v klidu i během přenosu.

Role-Based Access Control (RBAC): Jak efektivně používat RBAC a spravovat přístupová práva v prostředí Cosmos DB. [4,18]

### 8.4 Monitorování a ladění výkonu

Nástroje pro monitorování: Jak používat Azure Monitor, diagnostické logy a další nástroje pro sledování výkonu Cosmos DB.

Ladění výkonu: Identifikace a řešení problémů s výkonem, optimalizace databázových operací. [4,18]

### 8.5 Správa a škálování

Horizontální a vertikální škálování: Jak efektivně škálovat Cosmos DB, rozdíly mezi horizontálním a vertikálním škálováním.

Automatizace škálování: Jak nastavit automatické škálování na základě aktuální zátěže. [4,18]

## **8.6 Migrace dat a zálohování**

Migrace dat: Jak efektivně migrovat data do Cosmos DB z různých zdrojů.

Zálohování a obnova dat: Jak nastavit a spravovat zálohování, jak provádět obnovu dat v případě potřeby. [4,18]

## **8.7 Integrace s dalšími službami Azure**

Integrace s Azure Functions a Logic Apps: Jak propojit Cosmos DB s dalšími Azure službami pro automatizaci a rozšíření funkcionality.

Real-time aplikace: Příklady využití Cosmos DB v real-time aplikacích, jak zajistit rychlou a spolehlivou komunikaci. [4,18]

## ZÁVĚR

V dnešní době, kdy digitální technologie pronikají do všech aspektů našeho života, se zpracování a analýza dat stává klíčovou kompetencí pro mnoho odvětví. Jedním z přístupů, který nabízí řešení pro efektivní manipulaci s rozsáhlými objemy dat, je využití cloudových databázových technologií. Microsoft Azure Cosmos DB, představující multimodelovou databázovou platformu, se v tomto kontextu jeví jako inovativní nástroj schopný adresovat specifické požadavky aplikací vyžadujících vysokou dostupnost a globální distribuci dat.

Cílem této bakalářské práce bylo nejenom seznámení se s aktuální verzí Cosmos DB, ale především vytvoření podkladů, které by mohly sloužit pro výuku této moderní technologie. Práce zahrnovala tvorbu vzorových příkladů, cvičení a testovacích úkolů, které pomáhají studentům lépe porozumět a prakticky využívat Cosmos DB ve svých budoucích projektech.

Multimodelová schopnost Cosmos DB umožňuje vývojářům používat různé datové modely podle potřeb jejich aplikací, což činí tuto platformu extrémně flexibilní. V průběhu práce byly probrány klíčové koncepty jako globální distribuce, konzistence dat, modely konzistence a strategie škálování, které jsou zásadní pro optimalizaci výkonu a efektivity databázových operací v cloudovém prostředí.

Výsledky této práce poskytly podrobný přehled a praktické příklady použití Azure Cosmos DB, které mohou být integrovány do výukových materiálů. Studenti technických oborů tak získali cenné dovednosti v práci s cloudovými databázemi, což jim umožní efektivněji pracovat na svých projektech a lépe se připravit na budoucí kariéru v oblasti IT.

Navrhované možné směry rozvoje zahrnují integraci s jinými službami Azure, práci s reálnými datovými sadami nebo rozšíření o nové moduly zahrnující pokročilé techniky databázového managementu. Tyto rozvojové možnosti nejen že rozšíří současné znalosti studentů, ale také je připraví na budoucí výzvy a požadavky v oblasti datového managementu.

Na oficiálním blogu Microsoft Azure Scott Guthrie zdůraznil význam Azure Cosmos DB, zvláště v kontextu její schopnosti zpracovávat data na planetární úrovni. Uvedl, že Azure Cosmos DB je první globálně distribuovaná multimodelová databáze, která zajišťuje zpracování dat s garantovanou odezvou v milisekundách a vysokou dostupností. Tyto schopnosti, podle Guthrieho, otevírají nové možnosti pro vývojáře při vytváření

intelligentních aplikací, zvláště těch, které se škálují pro globální uživatele a integrují se s technologiemi umělé inteligence.

Tato bakalářská práce tedy nejen že přispěla k hlubšímu porozumění a praktickému využití Azure Cosmos DB, ale také položila základy pro další vývoj a inovace v oblasti výuky cloudových databázových technologií.

**SEZNAM POUŽITÉ LITERATURY**

- [1] GUTHRIE, Scott. Nové inovace na Microsoft Build 2017: Pomoc vývojářům dosáhnout více. Online. Bez verze. Redmond, WA. 2017. Dostupné z: <https://azure.microsoft.com/en-us/blog/new-innovations-at-microsoft-build-2017-helping-developers-achieve-more/>. [cit. 2024-04-15].
- [2] SHUKLA, Dharma. Azure Cosmos DB - Microsoftova globálně distribuovaná multimodelová databázová služba. Online. Bez verze. Redmond, WA. Dostupné z: <https://azure.microsoft.com/en-us/blog/azure-cosmos-db-microsofts-globally-distributed-multi-model-database-service/>. [cit. 2024-04-15].
- [3] MICROSOFT. Produkty Microsoft Azure. Online. Bez verze. Redmond, WA. Dostupné z: <https://azure.microsoft.com/en-us/products/>. [cit. 2024-04-15].
- [4] MICROSOFT. Dokumentace Microsoft Azure Cosmos DB. Online. Bez verze. Redmond, WA. Dostupné z: <https://learn.microsoft.com/en-us/azure/cosmos-db/>. [cit. 2024-04-15].
- [5] MICROSOFT. Vyberte si API pro Azure Cosmos DB. Online. Bez verze. Redmond, WA. Dostupné z: <https://learn.microsoft.com/en-us/azure/cosmos-db/choose-api>. [cit. 2024-04-15].
- [6] HATTIE, John; DONOGHUE, Gregory. Learning strategies: A synthesis and conceptual model. *Science of Learning*. 2016, vol. 1, no. 1, s. 1-13. Licence: CC BY. ISSN 2056-7936.
- [7] MOORE, Kenneth D. *Effective instructional strategies: from theory to practice*. 4th ed. Los Angeles: SAGE, c2015. ISBN 9781483306582.
- [8] FELDER, Richard M. a BRENT, Rebecca. *Teaching and learning STEM: a practical guide*. San Francisco: Jossey-Bass, A Wiley Brand, [2016]. ISBN 9781118925812.
- [9] ASANA. SMART Goals: Definition, Examples, and Template. [online]. Dostupné z: <https://asana.com/resources/smart-goals>. [cit. 2024-04-15].
- [10] NILSON, Linda B. a Ludwika A. GOODSON. *Online Teaching at Its Best: Emerging Instructional Design with Teaching and Learning Research*. 2nd ed. Hoboken, NJ: Wiley, 2020. ISBN 978-1119765011.
- [11] KOKOTSAKI, Dimitra, Andy WIGGINS a Victoria MENZIES. Project-based learning: A review of the literature. *Improving Schools*. 2016, 19(3), 267-277. Dostupné také z: doi:10.1177/1365480216659733



- [12] AKÇAYIR, Gökçe a Murat AKÇAYIR. The flipped classroom: A review of its advantages and challenges. *Computers & Education*. 2018, 126, 334-345. Dostupné taks: doi:10.1016/j.comp.edu.2018.07.021
- [13] KALIISA, Rogers, Bart RIENTIES, Anders I. MØRCH a Anders KLUGE. Social learning analytics in computer-supported collaborative learning environments: A systematic review of empirical studies. *Computers in Human Behavior*. 2022, 128, 100073. Dostupné také z: doi:10.1016/j.cb.2022.100073
- [14] ALHAZZANI, Noura. MOOC's impact on higher education. *Social Sciences & Humanities Open*. 2020, 2(1), článek 100030. Dostupné také z: doi:10.1016/j.ssaho.2020.100030
- [15] TAYLOR, Deborah L., YEUNG, Michelle a BASKET, A. Z. Personalized and Adaptive Learning. In: *Innovative Learning Environments in STEM Higher Education: Opportunities, Challenges, and Looking Forward*. [s.l.]: Springer Nature, 2021. Dostupné z: <https://library.oapen.org/bitstream/handle/20.500.12657/47325/1/9783030589486.pdf#page=31> [cit. 2024-04-17].
- [16] McDOWELL, Elizabeth a Kay SAMBELL. *Assessment and Feedback in Higher Education: A Guide for Teachers*. London: UCL Press, 2020. Dostupné tanks: <https://discovery.ucl.ac.uk/id/eprint/10096352/1/Assessment-and-feedback-in-higher-education.pdf>
- [17] MICROSOFT. NewDay revolutionizes the UK credit industry with Azure Cosmos DB. [online]. [cit. 14. května 2024]. Dostupné z: <https://customers.microsoft.com/en-us/story/1729598445861152322-newday-azure-cosmos-db-united-kingdom>
- [18] GUAY PAZ, José Rolando. *Microsoft Azure Cosmos DB Revealed*. 1. vyd. Berkeley, CA: Apress, 2018. ISBN 978-1-4842-3678-7.

## SEZNAM OBRÁZKŮ

Obrázek 1. Příklad slajdu z prezentaci. Proč Azure Cosmos DB? .....	28
Obrázek 2. Příklad slajdu z prezentaci. Přehled datových modelů v Cosmos DB. ....	30
Obrázek 3. Příklad slajdu z prezentaci. Modely konzistence v Azure Cosmos DB. ...	31
Obrázek 4. Příklad slajdu z prezentaci. Oddíly v Azure Cosmos DB. ....	32
Obrázek 5. Příklad slajdu z prezentaci. Výužití indexů pro zrychlení dotazů. ....	34
Obrázek 6. Příklad slajdu z prezentaci. Auditování a sledování přístupu. ....	35
Obrázek 7. Příklad slajdu z prezentaci. Nástroje pro monitorování a ladění.....	36
Obrázek 8. Příklad slajdu z prezentaci. Přehled integrace Azure služeb.....	37
Obrázek 9. Příklad slajdu z prezentaci. Nástroje pro migraci.....	38
Obrázek 10. Příklad slajdu z prezentaci. Case Study společnosti NewDay. ....	40
Obrázek 11. Příklad slajdu z prezentaci. Vytvoření účtu Azure.....	42
Obrázek 12. Příklad slajdu z prezentaci. Vytvoření účtu Cosmos DB. ....	43
Obrázek 13. Příklad slajdu z prezentaci. Vytvoření databáze a kontejneru. ....	44
Obrázek 14. Příklad slajdu z prezentaci. Konfigurace škálování. ....	45
Obrázek 15. Příklad slajdu z prezentaci. Instalace Node.js. ....	46
Obrázek 16. Příklad slajdu z prezentaci. Připojení k Azure Cosmos DB.....	47
Obrázek 17. Příklad slajdu z prezentaci. Funkce pro aktualizace dat.....	48
Obrázek 18. Příklad slajdu z prezentaci. Funkce pro mazání dat. ....	49
Obrázek 19. Příklad slajdu z prezentaci. Příklad čtení dat v Cosmos DB.....	50
Obrázek 20. Příklad slajdu z prezentaci. Kritéria hodnocení zápočtového projektu. .	51
Obrázek 21. Příklad webu pro správu databáze zákazníků pomocí Cosmos DB se základními CRUD operacemi. ....	66

## SEZNAM PŘÍLOH

- [1] 10 prezentací PowerPoint pro přednášky,
- [2] 10 prezentací PowerPoint pro cvičení,
- [3] Soubor JavaScript „index.js“,
- [4] Složka ZIP “customer-management“.