

Česká zemědělská univerzita v Praze

Technická fakulta

Katedra technologických zařízení staveb



Bakalářská práce

Porovnání vybraných databázových systémů

Autor: Zdeněk Prokop

Vedoucí bakalářské práce: Ing. Marek Pačes

Rok: © 2024 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Zdeněk Prokop

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Porovnání vybraných databázových systémů

Název anglicky

Comparison of selected database systems

Cíle práce

Cílem práce je porovnat min. 3 vybrané databázové systémy, zdůvodnit jejich vhodnost nasazení pro typické instalace, posoudit perspektivu dalšího vývoje, ověřit rychlost, stabilitu a technické požadavky pro chod systému.

Metodika

1. Analýza vybraných databázových systémů a dotazovacích jazyků
2. Explanace pojmů a parametrů databázových systémů
3. Komparace zjištěných dat
4. Indukce komparovaných dat
5. Abdukce perspektivy vývoje

Doporučený rozsah práce

30 až 40 stran textu včetně obrázků, grafů a tabulek

Klíčová slova

databáze, SQL

Doporučené zdroje informací

Lacko, L., Mistrovství v SQL Server 2012 – Kompletní průvodce databázového experta. Vyd. 1. Brno: CPress, 2013. ISBN 978-80-2513-773-4

Momjian, B., PostgreSQL: praktický průvodce. Computer Press, 2003. ISBN 978-80-7226-954-9

Welling, L., Thomson, L. MySQL Průvodce základy databázového systému. Vyd 1. Computer Press, 2005. ISBN 978-80-251-0671-6

Předběžný termín obhajoby

2023/2024 LS – TF

Vedoucí práce

Ing. Marek Pačes

Garantující pracoviště

Katedra technologických zařízení staveb

Elektronicky schváleno dne 29. 6. 2022

doc. Ing. Jan Malaťák, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 21. 9. 2022

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 13. 02. 2024

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Porovnání vybraných databázových systémů“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 25.3.2024

.....

podpis

Poděkování

Rád bych touto cestou poděkoval Ing. Markovi Pačesovi za vedení, rady a odbornou pomoc při psaní bakalářské práce.

Porovnání vybraných databázových systémů

Abstrakt

Bakalářská práce se zabývá problematikou databázových systémů. Teoretická část je věnována popisu tří databázových systémů, jejich vlastností a možností, a porovnání na základě společných parametrů. Následuje analýza zátěžových testů používaných pro porovnání relativních výkonů databází a popis principu zátěžového testu použitého v praktické části práce. V praktické části je poté popsána (a provedena) příprava a nezbytné kroky (instalace, konfigurace příslušných systémů a software, nastavení parametrů) předcházející samotnému měření, provedeno samotné měření a na základě výsledků tohoto měření, doporučeno konkrétní nasazení. V závěru práce je provedeno shrnutí jak praktické, tak teoretické části práce.

Klíčová slova: databáze, SQL, tabulka, transakce, zátěžový test

Comparison of selected database systems

Abstract

The bachelor's thesis deals with the issue of database systems. The theoretical part is devoted to the description of three database systems, their properties and possibilities, and comparison based on common parameters. This is followed by an analysis of the load tests used to compare the relative performances of the databases and a description of the principle of the load test used in the practical part of the work. In the practical part, the preparation and necessary steps (installation, configuration of relevant systems and software, parameter settings) preceding the actual measurement are described (and performed), the measurement itself is performed and, based on the results of this measurement, specific deployment is recommended. At the end of the work, a summary of both the practical and the theoretical part of the work is made.

Keywords: database, SQL, table, transaction, benchmark

Obsah

1	Úvod	1
2	Metodika a cíl práce	2
3	Popis použitých pojmů	3
3.1	Databáze.....	3
3.1.1	Relační model databáze (RDBMS).....	3
3.1.2	Hierarchický model databáze	4
3.1.3	Síťový model databáze	4
3.1.4	Objektový model databáze	5
3.1.5	SQL (Structured Query Language).....	5
3.1.6	Tabulka	5
3.1.7	Index.....	6
3.1.8	Pohled.....	6
3.1.9	Transakce	6
3.1.10	MVCC.....	6
3.1.11	OLTP	6
3.1.12	OLAP	6
3.2	Aplikační logika (vícevrstvá architektura)	7
3.3	Licencování.....	7
3.3.1	EULA	7
3.3.2	Open-source	8
3.4	Bezpečnostní protokoly.....	8
3.4.1	SSL	8
3.4.2	TLS	8
3.5	Datový sklad	8
3.6	Benchmark	8
4	Přehled porovnávaných databázových systémů.....	9
4.1	PostgreSQL	9
4.1.1	Verze.....	9
4.1.2	Upgrade	9
4.1.3	Systémové požadavky	10
4.2	Microsoft SQL Server.....	10
4.2.1	Verze.....	10
4.2.2	Systémové požadavky	11
4.2.3	Edice	12
4.3	MySQL	12

4.3.1	Edice	13
4.3.2	Verze.....	13
4.3.3	Systémové požadavky	14
5	Vybrané srovnávací atributy	15
5.1	Licencování.....	15
5.2	Snadnost použití.....	15
5.3	Syntaxe	16
5.4	Datové typy	16
5.5	Geografická data	17
5.6	Rozlišování malých a velkých písmen (case sensitive)	17
5.7	Indexy	18
5.8	Replikace	18
5.9	Clustering	19
5.10	Vysoká dostupnost.....	19
5.11	Views	19
5.12	Triggery.....	20
5.13	Fulltextové vyhledávání.....	20
5.14	Celá čísla.....	21
5.15	Booleovské typy	22
5.16	NoSQL podpora	22
5.17	Bezpečnost	22
5.18	Nástroje pro správu – administrace pomocí GUI	22
5.19	Souběžnost.....	23
5.20	Výkon.....	23
5.21	Oblíbenost.....	24
5.22	Souhrn	25
6	Zátěžové testy	26
6.1	Syntetické benchmarky	26
6.2	Srovnávací testy v reálném světě.....	26
6.3	Důvody pro srovnávání databáze.....	26
6.4	TPC-A	27
6.5	TPC-C	27
6.6	Model TPC-C.....	27
6.7	HammerDB	28
6.8	TPROC-C	29
7	Praktická část.....	30

7.1	Testovací hardware a software	30
7.2	Napojení PostgreSQL.....	30
7.3	Napojení SQL Server.....	31
7.4	Napojení MySQL.....	32
7.5	Uložení konfigurace.....	33
7.6	Spuštění a automatizace testů	33
7.7	Vlastní měření	34
7.8	Vyhodnocení.....	36
8	Závěr.....	39

Seznam obrázků

Obr. 1 – Hierarchický model databáze	4
Obr. 2 – Síťový model databáze	5
Obr. 3 – Třívrstvá architektura	7
Obr. 4 – Prostředí software HammerDB	29
Obr. 5 – Nastavení připojení k PostgreSQL	31
Obr. 6 – Grafické rozhraní pgAdmin s vytvořenou testovací databází.....	31
Obr. 7 – Správce zdrojů dat ODBC.....	32
Obr. 8 – Nastavení připojení k SQL Serveru	32
Obr. 9 – Nastavení připojení k MySQL	33
Obr. 10 – Výsledky počtu objednávek a počtu transakcí MySQL při 9. měření	34
Obr. 11 – Graf transakcí testu PostgreSQL při 9. měření.....	34
Obr. 12 – Souhrn 10. měření zobrazený ve webovém prohlížeči	35

Seznam tabulek a grafů

Tab. 1 – Hardwarové požadavky pro PostgreSQL 16.1	10
Tab. 2 – Přehled verzí SQL server	11
Tab. 3 – Hardwarové požadavky pro SQL server 2022	11
Tab. 4 – Přehled verzí MySQL	13
Tab. 5 - Hardwarové požadavky pro MySQL 8.2	14
Tab. 6 – Rozdíly syntaxe SQL mezi PostreSQL a SQL Server	16
Tab. 7 – Používané datové typy	16
Tab. 8 – Celá čísla v PostgreSQL	21
Tab. 9 – Celá čísla v SQL Server	21
Tab. 10 – Celá čísla v MySQL	21
Tab. 11 – Přehled srovnávacích kritérií	25
Tab. 12 – Parametry testovaného serveru.....	30
Tab. 13 - Aritmetický průměr (počet zpracovaných) OPM a TPM jednotlivých systémů všech 10. provedených měření	36
Tab. 14 - Aritmetický průměr OPM a TPM, rozšíření Tab. 13 o SQL Express	37

Graf 1 – Počet OPM pro jednotlivé systémy (aritmetický průměr všech 10. provedených měření)	36
Graf 2 – Počet TPM pro jednotlivé systémy (aritmetický průměr všech 10. provedených měření).....	36
Graf 3 – Počet zpracovaných OPM, rozšíření o SQL Express.....	37
Graf 4 – Počet zpracovaných TPM, rozšíření o SQL Express	38

1 Úvod

Databáze byly, jsou a budou součástí každého software, a to nejen takového, u kterého se čtení a ukládání dat předpokládá, jako např. program používaný v knihovnách pro evidenci knih, jejich výpůjček a vrácení, a evidenci registrovaných čtenářů. Ale i u jednoduchých aplikací, které si „umí“ zapamatovat např. poslední hledanou nebo vypočtenou hodnotu, i když na pozadí takové aplikace nemusí být zrovna celý databázový systém, pouze textový soubor, do kterého se ukládají data ve formě strukturovaných řádků s oddělovači.

V dnešní době si už ani neumíme představit život bez mobilního telefonu a aplikací v něm nainstalovaných. Téměř každý člověk, a nejen mladý, v České republice zná nebo je součástí nějaké sociální sítě. Většina lidí, kteří cestují veřejnou dopravou, využívá on-line jízdní řády a jízdenky nakupuje právě pomocí takovýchto aplikací (což jsou vlastně specializované e-shopy, jen místo zboží prodávají služby). Všichni studenti (a pedagogové) naší univerzity používají UIS (Univerzitní Informační Systém), ve kterém je uloženo velké množství studijních dat, od jmen a fotografií jednotlivých studentů, výsledků jejich studia, přes přihlašování (a odhlašování) se na zkoušky, rozvrhů jednotlivých studijních oborů a ročníků, kapacity jednotlivých poslucháren a učeben, až po archiv bakalářských a diplomových prací. I běžnému uživateli informačních technologií je jasné, že uložení takového množství dat musí být provedeno pomocí nějakého sofistikovaného nástroje, který pracuje na pozadí uživatelského rozhraní (tzv. frontendu). A tímto nástrojem jsou právě databáze a databázové systémy, které data uchovávají, umožňují jejich načtení a editaci. V případě havárie hardware, na kterém takový systém „běží“, umožní systém rychlou obnovu do stavu před výpadkem (dnes i v reálném čase), takže běžný uživatel ani žádný takový výpadek nezaznamená, a nedojde ani ke ztrátě právě editovaných dat.

Věřím, že i čtenáři bez hlubších znalostí v oblasti informačních technologií, poskytnou tato práce min. základní přehled o databázových systémech a možnostech použití.

2 Metodika a cíl práce

V teoretické části budou analyzovány vlastnosti databázových systémů, možnosti použití a rozšíření pro oblast vysoké dostupnosti a replikace dat, a jejich schopnost zpracovat dotazy dotazovacího jazyka SQL a základní i rozšířené datové typy.

V praktické části práce bude porovnán relativní výkon vybraných databázových systémů pomocí srovnávacího testu TPC-C. Pro srovnávací test bude použit virtuální server (technologie Hyper-V společnosti Microsoft), připojený v počítačové síti, s nainstalovaným OS Microsoft Windows Server 2022 Standard. Konkrétní hardwarové parametry serveru budou uvedeny v praktické části. Na serveru budou nainstalovány databázové systémy Microsoft SQL Server 2022 Standard Edition, PostgreSQL 16.1 a MySQL 8.2. Aby test na daném databázovém systému nebyl ovlivněn během ostatních dvou systémů, budou vždy zbylé dva systémy vypnuté (zastavena příslušná služba). Instalace databázových systémů bude provedena pomocí tzv. MSI balíčků (standardní instalátory pro OS Windows) a budou ponechány defaultní konfigurace (mimo nastavení hesel administrátorských účtů).

Pro porovnání výkonů jednotlivých databázových systémů bude využit software HammerDB 4.9 nainstalovaný na běžné klientské stanici stejné sítě. Nastavení parametrů testu (počet virtuálních uživatelů, časy testu, počet prováděných transakcí apod.) bude pro porovnávané databázové systémy stejné. Tento software simuluje společnost, která prodává zboží, naskladňuje jej, přesouvá zásoby mezi sklady a vyřizuje objednávky zákazníků. Bližší popis modelu tohoto testu je popsán v kapitole 6.6. Test pro každý systém bude opakován desetkrát v různých časech a dnech. Testy by měly být provedeny při běžném zatížení sítě, na které bude probíhat i další provoz běžně používaných systémů.

Výsledky (počet provedených transakcí a počet provedených objednávek za minutu) budou zapisovány do excelovských tabulek, které budou publikovány v praktické části a následně použity pro komparaci zjištěných dat a jejich interpretaci pro vhodnost nasazení v typických instalacích.

Cílem práce je tedy popis nejpoužívanějších databázových systémů, které „pohání“ aplikace zpracovávající data. Práce poskytne porovnání vybraných parametrů těchto systémů pro uchování a zpracování „velkých“ dat. Praktická část práce bude porovnávat tři vybrané databázové systémy pomocí benchmarkového (srovnávacího) a zátěžového testu, který v každém systému vytvoří databázi fiktivní společnosti, spolu s příslušnými tabulkami pro artikly, zákazníky, objednávky a sklady. Následně tento zátěžový test bude vyřizovat objednávky z příslušných skladů a výsledkem porovnání bude počet transakcí a objednávek, který je daný databázový systém schopen zpracovat za minutu.

K dosažení stanoveného cíle budou v teoretické části práce zpracovány především informace ze zdrojů na webových stránkách dodavatelů nebo distributorů databázových systémů a z odborné literatury a dále popis použitých pojmů ze stránek zabývajících se informačními technologiemi. Pro praktickou část, ve které bude provedeno měření výkonů popisovaných databázových systémů, bude využit software HammerDB, který simuluje výše popsáný zátěžový test.

3 Popis použitých pojmů

V kontextu uchování dat se pojem databáze užívá od konce šedesátých let 20. století, s rozvojem potřeby snazší správy, stále větších dat, o občanech státu. Prvopočátky databází je možné vidět v antice, která už pracovala s archivy, tedy místy, kde se shromažďovaly úřední dokumenty. Přímým předchůdcem pak je třeba papírová kartotéka. Ta umožňovala uspořádání dat podle různých kritérií a zařídování nových položek. Veškeré operace s nimi prováděl přímo člověk. Správa takových kartoték byla v mnohém podobná správě dnešních databází. Dalším krokem bylo převedení zpracování dat na stroje. Za první velké strojové zpracování dat lze asi považovat sčítání lidu ve Spojených státech v roce 1890. Paměťovým médiem byl dřevěný štítek a zpracování sebraných informací probíhalo na elektromechanických strojích. Elektromechanické stroje se využívaly pro účely zpracování dat další půlstoletí. Velkým impulsem pro další rozvoj databází byl překotný vývoj počítačů v padesátých letech 20. století. V roce 1959 se konala konference zástupců firem, uživatelů a amerického ministerstva obrany, jejímž záměrem byl požadavek na univerzální databázový jazyk. Výsledkem byla o 4 roky později na konferenci CODASYL publikovaná první verze jazyka COBOL, který byl po mnoho dalších let nejrozšířenějším jazykem pro hromadné zpracování dat. S rozvojem technologií se rozvíjely také databáze.[1]

3.1 Databáze

Databáze je nástroj pro shromažďování a uspořádávání informací. Do databází se ukládají informace např. o osobách, produktech, objednávkách nebo čemkoliv jiném. Původ mnohých databází je v tabulkových procesorech nebo jako seznam uložený v textovém souboru. Jak se jednotlivé seznamy zvětšují, začínou se v uložených datech objevovat redundance a nekonzistence. Tím už se data ve formě seznamu stávají nepřehledná a hůře vyhledatelná, a možnosti, jak je právě vyhledat nebo si z nich vybrat určitou část, jsou omezené. Jakmile se tyto problémy začnou objevovat, je nejvhodnější čas data převést do databáze, vytvořené pomocí systému pro správu databází.

Počítačová databáze je kontejner objektů. Jedna databáze téměř vždy obsahuje více než jednu tabulku. Například systém sledování skladových zásob, který využívá tři tabulky, nejsou tři databáze, ale jedna databáze, která obsahuje tři tabulky.[2] Kromě tabulek obsahující data jsou v databázích i další objekty jako pohledy (views), vložené procedury a další.

Nejpoužívanějším typem je v současnosti relační databáze. Existují i další typy jako hierarchická, síťová a objektová databáze.[3]

3.1.1 Relační model databáze (RDBMS)

Jejich základem jsou tzv. relace (propojení předem nastavenými vztahy mezi tabulkami). Tabulka je základní objekt každé databáze a obsahuje jednotlivé sloupce – **atributy** a řádky – **záznamy**. Atributy tabulky určují vlastnosti údajů, které se do tabulky budou vkládat. Do tabulky se vkládají údaje stejného druhu (např. kód, název a cena skladovaného zboží, nicméně každý jen jednou z důvodu neredundance). Atributy při návrhu tabulek neobsahují samotné hodnoty, pouze určují, jaké vlastnosti budou vkládané údaje mít.

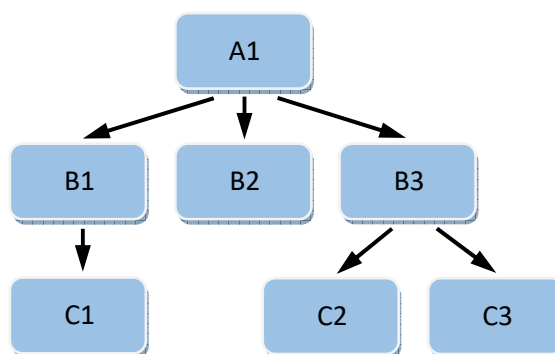
Existují speciální druhy atributů, které plní v relační databázi specifické funkce. Je to **primární klíč** – tabulky nemají pevně zafixované pořadí záznamů (řádků), a pro jednoznačnou identifikaci daného záznamu (např. pro editaci nebo jeho výběr), je nutné určit jeden z atributů tabulky jako unikátní. Tento atribut se nazývá primární klíč a každý záznam v tabulce musí mít hodnotu tohoto atributu unikátní pro celou tabulku (dva lidé mohou mít stejné jméno a příjmení, ale ne stejné RČ).

Každá tabulka musí primární klíč obsahovat. Druhým speciálním atributem je **cizí klíč** – záznamy v tabulkách, které jsou propojeny relacemi musí být nějakým způsobem provázány. Provázání se docílí vytvořením cizího klíče v tabulce. Do tohoto atributu se pak bude u jednotlivých záznamů kopírovat hodnota primárního klíče navázaného záznamu z druhé tabulky.[4]

3.1.2 Hierarchický model databáze

Jedná se o jeden z nejstarších modelů v datovém modelu, který vyvinula firma IBM v 50. letech 20. století. V hierarchickém modelu jsou data uspořádána jako kolekce tabulek nebo segmenty, které spolu tvoří hierarchický vztah. V tomto modelu jsou data organizována do stromové struktury, kde každý záznam sestává z jednoho rodičovského záznamu a mnoha potomků. I když jsou segmenty propojeny jako řetězová struktura logickými asociacemi, pak struktura může být vějířová struktura s více větvemi. Nelogické asociace nazýváme asociacemi směrovými.

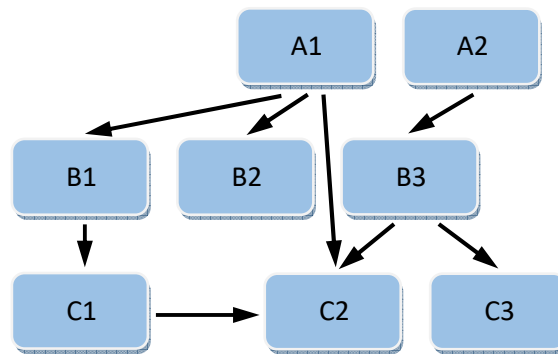
V hierarchickém modelu (viz Obr. 1) se segmenty, na které ukazuje logická asociace, nazývají podřízený segment a druhý segment se nazývá nadřazený segment. Pokud existuje segment bez rodiče, pak se bude nazývat kořen a segmenty, které nemají žádné potomky, se nazývá listy. Hlavní nevýhodou hierarchického modelu je, že mezi uzly mohou být vztahy 1:1 a 1:N.[5]



Obr. 1 – Hierarchický model databáze

3.1.3 Síťový model databáze

Tento model byl formalizován skupinou Database Task v 60. letech 20. století. Tento model je zobecněním hierarchického modelu. Tento model se může skládat z více nadřazených segmentů a tyto segmenty jsou seskupeny jako úrovně, ale mezi segmenty patřícími do jakékoli úrovně existuje logické spojení. Většinou existuje logická asociace many-to-many mezi kterýmkoliv ze dvou segmentů. Grafy se pak nazývají logickými asociacemi mezi segmenty. Proto tento model nahrazuje hierarchický strom strukturou podobnou grafu a díky tomu mohou existovat obecnější vazby mezi různými uzly. Může mít vztahy M:N (viz Obr. 2), tj. many-to-many, což umožňuje, aby záznam měl více než jeden nadřazený segment.[6]



Obr. 2 – Síťový model databáze

3.1.4 Objektový model databáze

Jedná se o databázový model, kde jsou data reprezentována ve formě objektů, jako se používají v objektově orientovaném programování. Objektový model implementuje objektově orientované koncepty, jako jsou třídy objektů, identita objektu, polymorfismus, zapouzdření a dědičnost. Objektově orientovaná databáze ukládá, ve srovnání s relační databází, komplexní data.[7]

3.1.5 SQL (Structured Query Language)

SQL je standardizovaný strukturovaný dotazovací jazyk, který se používá ke správě relačních databází (RDBMS) a operací se záznamy (přidávání, editace a mazání) v těchto databázích. SQL byl původně vytvořen v 70. letech 20. století a používají jej nejen správci databází, ale také vývojáři píší skripty pro integraci dat a analytici dat pro spuštění analytických dotazů nad příslušnými daty.

Jako programovací jazyk má SQL příkazy a syntaxi pro provádění těchto příkazů. Tyto příkazy jsou rozděleny do několika skupin:

- Příkazy jazyka **DDL (Data Definition Language)** – příkazy definice dat, používané k definování datových tabulek.
- Příkazy jazyka **DML (Data Manipulation Language)** – příkazy pro manipulaci dat v definovaných datových tabulkách. Příkazy **INSERT**, **UPDATE** a **DELETE** pro vložení, změnu a smazání záznamů. Na rozdíl od příkazů DDL, které definují způsob ukládání dat, příkazy DML fungují v tabulkách definovaných pomocí příkazů DDL.
- **DQL (Data Query Language)** – je pouze jediný příkaz, **SELECT**, který se používá k získání konkrétních dat z tabulek. Tento příkaz je obvykle seskupen s příkazy DML.
- Příkazy jazyka **DCL (Data Control Language)** – příkazy pro nastavování (**GRANT**) a odebírání (**REVOKE**) přístupových oprávnění a k ovládání transakcí, kde **BEGIN** transakci zahájí a **COMMIT** ji potvrdí nebo **ROLLBACK** ji zruší.
- Ostatní příkazy – do této skupiny patří zejména příkazy, které slouží k popisu doplňujících objektů databázových systémů, určené zejména pro popis pohledů (views), uložených procedur, funkcí a triggerů, což jsou de facto podprogramy (aplikační logika) běžící na úrovni databázového serveru.

[8]

3.1.6 Tabulka

Tabulka je jedním ze základních databázových objektů, který slouží k přímému uložení dat do paměťového prostoru relační databáze. Databázovou tabulku je možné si představit jako běžnou dvourozměrnou tabulku s pevně daným počtem a významem jednotlivých položek, které tvoří

sloupce tabulky. Záznamy jsou do tabulky ukládány ve formě řádků. Každý záznam je tak tvořen stejným počtem položek a stejného datového typu v daném sloupci. Není proto možné, aby dva různé záznamy ve stejné tabulce měly rozdílný počet položek nebo obsahovaly ve stejné položce rozdílné datové typy. Počet záznamů (řádků) v tabulce je obvykle omezen jen technickými možnostmi použité databáze.[9]

3.1.7 Index

Databázové indexy slouží ke zrychlení přístupu k datům a měly by se používat u všech sloupců, podle kterých se vyhledává, třídí nebo podle kterých se spojují tabulky.

Při ukládání dat do tabulek nejsou záznamy obvykle nijak tříděny a ukládají se většinou za sebe tak, jak byly postupně vloženy. V momentě, kdy chceme data z tabulky později vybrat podle nějakého kritéria, je nutné projít všechny záznamy a vybrat z nich ty, které kritériu vyhovují. K tomu, aby při výběru několika záznamů nebylo potřeba procházet všechny ostatní, slouží právě indexy, ve kterých jsou data organizována tak, aby bylo možné rychle vybrat pouze relevantní záznamy.

Indexy se vytvářejí nad jedním nebo několika sloupci tabulky, každá tabulka může mít indexů několik. Index definovaný nad sloupcem tabulky umožňuje rychlý přístup k záznamům podle hodnot tohoto sloupce.[10]

3.1.8 Pohled

Pohled neboli view, je databázový objekt, který poskytuje data ve stejné podobě jako tabulka. Na rozdíl od tabulky, kde jsou data přímo uložena, obsahuje pohled pouze předpis (vytvořený pomocí jazyka DDL), jakým způsobem a z kterých tabulek nebo i jiných pohledů mají být data získána.[11]

3.1.9 Transakce

Je logická množina nebo sekvence operací, která do této transakce patří. Používají se při provádění změn nad záznamy (**INSERT**, **UPDATE**) v databázích. Výhodou používání transakcí je, že provedené změny jsou odvolatelné a transakci je nutné vždy potvrdit příkazem **COMMIT** nebo odvolat pomocí příkazu **ROLLBACK**. Zejména při výskytu chyby při provádění více operací je složité dohledat, které změny již proběhly a které ne, proto zejména odvolání celé transakce, vrátí již změněné záznamy do původní podoby.[12]

3.1.10 MVCC

Multiversion concurrency control – technika optimalizace databáze, která vytváří duplicitní kopie záznamů, takže data lze bezpečně číst a aktualizovat současně, a tím se čtení a zápis navzájem neblokují.[13]

3.1.11 OLTP

Online Transaction Processing je typ zpracování dat, který sestává z provádění řady transakcí probíhajících současně – například online bankovníctví, nakupování, zadávání objednávek nebo odesílání textových zpráv. Tyto transakce se tradičně označují jako ekonomické nebo finanční transakce, zaznamenané a zabezpečené tak, aby uživatelé měli kdykoli přístup k informacím pro účely účetnictví nebo výkaznictví.[14]

3.1.12 OLAP

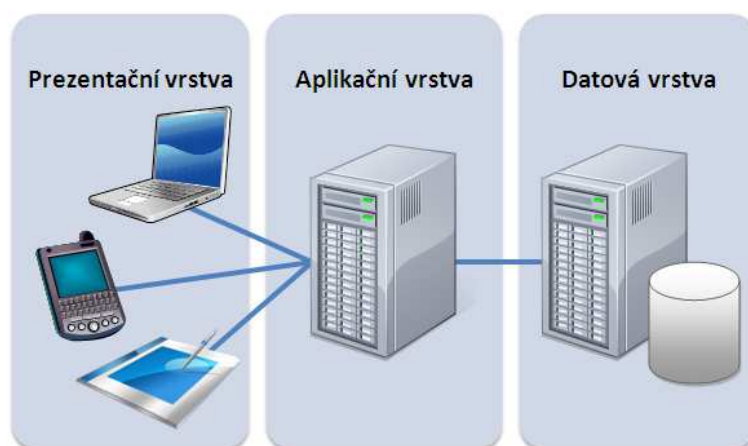
Online Analytical Processing je databáze usnadňující dotazy business intelligence (extrahování dat z databáze a jejich analýza pro informované obchodní rozhodnutí). Tato databázová technologie je optimalizovaná pro dotazování a vytváření sestav místo zpracování transakcí. Zdrojová data pro OLAP jsou databáze OLTP, které jsou běžně uložené v datových skladech. Data OLAP jsou odvozená

z historických dat a agregována do struktur, které umožňují sofistikovanou analýzu. Data OLAP jsou také uspořádaná hierarchicky a uložena v tzv. kostkách místo tabulek. Je to sofistikovaná technologie, která využívá multidimenzionální struktury k zajištění rychlého přístupu k datům pro analýzu.[15]

3.2 Aplikační logika (vícevrstvá architektura)

Vícevrstvá architektura (viz Obr. 3) označuje jeden z typů architektury informačních systémů (resp. aplikací). Tedy to, jakým způsobem je aplikace rozdělena mezi to, co vidí a používá uživatel (tzv. prezentační vrstva) a to, co se odehrává na pozadí na straně serverů (aplikační a datová vrstva).

- **Prezentační vrstva** – část, která je „viditelná“ pro uživatele (frontend), zajišťuje vstup požadavků a prezentaci výsledků. Je závislá na platformě (např. webová aplikace, aplikace pro systém Windows, Android aplikace apod.). Může být tedy různá pro různá zařízení či platformy.
- **Aplikační vrstva** (funkční) – prostřední vrstva vícevrstvého modelu (middleware), zajišťuje výpočty a operace prováděné mezi vstupně-výstupními požadavky a daty. Je také nazývána jako aplikační server.
- **Datová vrstva** (databázová) – nejnižší vrstva modelu, zajišťuje práci s daty, tedy systém řízení báze dat a základní datově-funkční operace zajišťující ukládání, výběr, agregaci, předzpracování a integritu dat.



Obr. 3 – Třívrstvá architektura [16]

Výhodou vícevrstvé architektury je oddělení jednotlivých vrstev tak, aby na sobě nebyly závislé. Vícevrstvou architekturu využívá velké množství aplikací, které pracují s daty. Takto je postavena většina moderních podnikových aplikací, portálová řešení a webové stránky. Výhodou je pružnější rozdělení výkonu mezi zařízení uživatele a server, prezentační vrstva může běžet i na málo výkonných a levných zařízeních.[17]

3.3 Licencování

3.3.1 EULA

End User License Agreement – velmi striktně vymezuje, co uživatel může, a především nemůže, s takovým softwarem dělat. V každém případě nemůže (nejen z licenčních důvodů, ale už vzhledem k nedostupnosti zdrojových kódů) program upravovat a takto upravený ho dále šířit.[18]

3.3.2 Open-source

Je software se zdrojovým kódem, který může kdokoli kontrolovat, upravovat a vylepšovat. Zdrojový kód je část software, kterou většina uživatelů počítačů nikdy nevidí. Je to kód, který mohou počítačovní programátoři upravovat, aby změnili způsob, jakým program nebo aplikace funguje. Programátoři, kteří mají přístup ke zdrojovému kódu, mohou program nebo aplikaci vylepšit přidáním funkcí nebo opravou částí, které nefungují vždy správně.[19]

3.4 Bezpečnostní protokoly

3.4.1 SSL

Secure Sockets Layer – protokol, který zajišťuje bezpečnou komunikaci přes internet. Používá jak symetrickou, tak asymetrickou kryptografii. Tento protokol zajišťuje ověřování serveru a ověřování klientů:

- ověření serveru se provádí po připojení klienta k serveru. Po úvodním handshake (nastavení komunikačního kanálu), server odešle svůj digitální certifikát klientovi. Klient tento certifikát (nebo řetězec certifikátů) následně ověří.
- ověření klienta se provádí, po odeslání certifikátu serverem na klienta během handshake. Pokud je klientský certifikát nebo řetězec ověřen a je ověřena i zpráva o ověření certifikátu, proces handshake pokračuje.
- volitelné dodatečné ověření se provádí kontrolou běžného názvu v certifikátu proti plně kvalifikovanému názvu domény serveru z reverzního vyhledávání serveru DNS (Domain Name Server), kde lze získat plně kvalifikovaný název domény serveru.

[20]

3.4.2 TLS

Transport Layer Security – nahrazuje výše uvedený protokol SSL.

3.5 Datový sklad

Datový sklad (EDW) je podniková datová platforma používaná pro analýzu a vykazování strukturovaných a polostrukturovaných dat z různých zdrojů dat, jako jsou transakce v místě prodeje, marketingová automatizace, zákaznická řízení vztahů a další.

Datové sklady zahrnují analytickou databázi a kritické analytické komponenty a postupy. Podporují ad hoc analýzu a vlastní reporty. Mohou konsolidovat a integrovat obrovské množství aktuálních a historických dat na jednom místě a jsou navrženy tak, aby poskytovaly dlouhodobý pohled na data v průběhu času. Tyto funkce datového skladu učinily z datových skladů primární základ podnikových analýz, které pomáhají podporovat informovaná obchodní rozhodnutí.[21]

3.6 Benchmark

Benchmark (srovnávací test) je dobře definovaná a osvědčená technika pro porovnávání a analýzu výkonu databází nebo systémů správy databází (DBMS). Správci pomocí benchmarků sledují chování hardware, vymezují charakteristiky databáze a zjišťují, jak aplikace a databáze v praxi spolupracují.[22]

4 Přehled porovnávaných databázových systémů

Tato kapitola stručně popisuje jednotlivé porovnávané databázové systémy (výrobce, historie, vývoj), dále jsou zde uvedeny vydané verze, minimální systémové požadavky pro provoz na OS Windows a případně další informace.

4.1 PostgreSQL

PostgreSQL je pokročilý relační databázový systém s otevřeným zdrojovým kódem (tzv. open-source). Tento systém podporuje jak SQL (relační), tak JSON (nerelační) dotazování. PostgreSQL je vysoce stabilní a bezpečný systém podporovaný více než 20 lety vývoje open-source komunitou. Databáze PostgreSQL používá mnoho webových aplikací, ale i mobilní a analytické aplikace.

Projekt PostgreSQL začal v roce 1986 na Berkeley Computer Science Department, University of California. Projekt byl původně pojmenován Postgres jako odkaz na starší systém Ingres, který se také vyvíjel v Berkeley. Cílem projektu Postgres bylo přidat minimální funkce potřebné pro podporu více datových typů. V roce 1996 byl projekt Postgres přejmenován na PostgreSQL, aby byla jasně dána jeho podpora pro SQL. Dnes se PostgreSQL, stejně jako na začátku, označuje Postgres, ale dnes je to zkratka. Od té doby PostgreSQL Global Development Group (specializovaná komunita přispěvatelů) pokračuje ve vydávání open-source a bezplatného databázového projektu.[23]

Původně byl PostgreSQL navržen tak, aby běžel na platformách podobných UNIXu a následně byl PostgreSQL vyvinut pro různé platformy jako jsou Windows, macOS a Solaris.[24]

4.1.1 Verze

Nejnovější verze PostgreSQL 16 byla vydána 14.9.2023. PostgreSQL Global Development Group vydává novou tzv. hlavní (major) verzi, která obsahuje nové funkce přibližně jednou ročně. Opravy chyb a případné opravy zabezpečení jsou vydávány alespoň jednou za tři měsíce v tzv. dílčí (minor) verzi. Pokud vývojový tým usoudí, že kritická chyba nebo bezpečnostní oprava je příliš důležitá na to, aby „počkala“ na pravidelně plánované dílčí vydání, může zpřístupnit i vydání mimo plán těchto dílčích vydání.

PostgreSQL Global Development Group podporuje hlavní verzi po dobu 5 let od jejího prvního vydání. Po tomto 5letém období bude mít hlavní verze poslední dílčí verzi obsahující všechny opravy a bude považována za ukončenou a již nebude podporována.

Počínaje PostgreSQL 10 je hlavní verze označena zvýšením čísla první části verze, např. z 10 na 11. Před PostgreSQL 10 byla hlavní verze označena zvýšením buď první nebo druhé části čísla verze, např. 9.5 na 9.6. Dílčí verze jsou číslovány zvýšením čísla poslední části čísla verze. Počínaje PostgreSQL 10 je to druhá část čísla verze, např. 10.0 na 10.1 a u starších verzí je to třetí část čísla verze, např. 9.5.3 na 9.5.4. Je doporučeno, aby uživatelé instalovali nejnovější dílčí verzi pro jimi používanou hlavní verzi.[25]

4.1.2 Upgrade

Hlavní verze obvykle obsahuje změnu vnitřního formátu systémových tabulek a datových souborů. Tyto změny jsou často složité a není zachována zpětná kompatibilita uložených dat. Pro přechod (upgrade) na vyšší hlavní verzi, je většinou nutné provést zálohu a restore všech databází pomocí modulů `pg_dump` a `pg_restore` (ty jsou součástí instalace) nebo použití modulu `pg_upgrade`. Jestli je možné provést přechod z jedné hlavní verze na druhou, bez použití mezilehlých verzí, je nutné prostudovat si dokumentaci k jednotlivým verzím.

Upgrade dílčích verzí normálně backup a restore databází nevyžaduje, stačí jen zastavit službu databázového serveru, nainstalovat aktualizované binární soubory a službu databázového serveru opět spustit. U některých verzí mohou být k dokončení upgradu vyžadovány ruční změny, proto je opět nutné si prostudovat poznámky k verzi. Každý upgrade bude vždy obsahovat určitou úroveň rizika nicméně komunita považuje neprovádění upgradu za riskantnější než upgrade.[25]

4.1.3 Systémové požadavky

Níže v Tab. 1 je uveden přehled základních hardwarových požadavků pro provoz databázového systému PostgreSQL 16.1.

Tab. 1 – Hardwarové požadavky pro PostgreSQL 16.1 [26]

Komponenta	Požadavek
Pevný disk	Minimálně 512 MB volného místa na pevném disku. Pro data nebo podpůrné součásti je vyžadováno další místo na disku.
Paměť	Minimum: 2 GB.
Rychlost procesoru	Minimum: 1 GHz Dual Core.

4.2 Microsoft SQL Server

Microsoft SQL Server je databázový systém pro správu relačních databází vyvinutý společností Microsoft. Jedná o softwarový produkt s primární funkcí ukládání a získávání dat podle požadavků aplikací, které mohou běžet buď na stejném počítači (dnes již většinou serveru), nebo na jiném počítači v síti (včetně internetu). Na trhu je mnoho různých verzí a edic tohoto databázového systému, určených pro různé cílové skupiny a různé pracovní využití, od malých aplikací pro jeden počítač až po velké internetové aplikace pro mnoho uživatelů pracujících souběžně.

Od verze SQL Server 2016 je systém podporován pouze na procesorech x64 a musí mít minimálně 1,4 GHz procesor (doporučuje se 2,0 GHz nebo rychlejší).[27]

Od verze SQL Server 2017 je možné systém provozovat i na jiných operačních systémech než Microsoft Windows a to Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES) a Ubuntu. Je podporován také jako image kontejneru, která může běžet na Kubernetes, OpenShift a Docker Engine v Linuxu. Verze SQL Server 2022 je poslední vydanou verzí (16. listopadu 2022).[28]

4.2.1 Verze

Tab. 2 uvádí stručný přehled vydaných (hlavních) verzí SQL Server vč. číselného označení verzí, posledního vydaného servisního balíčku (dílčí verze), GDR (kumulativní aktualizace pro příslušnou dílčí verzi nebo servisní balíček a CU (souhrnné aktualizace pro danou hlavní verzi).

Tab. 2 – Přehled verzí SQL server [29]

Verze	Service pack	GDR	Cumulative update
SQL Server 2022 (16.0)	není	GDR (16.0.1050.5 – únor 2023)	CU8 for 2022 (16.0.4075.1 – září 2023)
SQL Server 2019 (15.0)	není	GDR (15.0.2101.7 – únor 2023)	CU22 for 2019 (15.0.4322.2 – srpen 2023)
SQL Server 2017 (14.0)	není	GDR (14.0.2047.8 – únor 2023)	CU31 for 2017 (14.0.3456.2 – září 2022)
SQL Server 2016 (13.0)	Azure Connect pack (13.0.7000.253 – květen 2022)	GDR for Azure Connect pack (13.0.7024.30 – únor 2023) GDR for RTM (13.0.1745.2 – leden 2018)	CU17 + GDR for SP2 (13.0.5893.48 – červen 2022)
SQL Server 2014 (12.0)	SP3 (12.0.6024.0 – říjen 2018)	GDR for SP3 (12.0.6174.8 – únor 2023)	CU4 + GDR for SP3 (12.0.6444.4 – únor 2023)
SQL Server 2012 (11.0)	SP4 (11.0.7001.0 – září 2017)	GDR for SP4 (11.0.7512.11 – únor 2023)	CU10 for SP3 (11.0.6607.3 – srpen 2017)
SQL Server 2008 R2 (10.50)	SP3 (10.50.6000.34 – září 2014)	GDR for SP3 (10.50.6785.2 – únor 2023)	není
SQL Server 2008 (10.0)	SP4 (10.0.6000.29 - září 2014)	GDR for SP4 (10.0.6814.4 - únor 2023)	není

4.2.2 Systémové požadavky

Níže v Tab. 3 je uveden přehled základních hardwarových požadavků pro provoz databázového systému SQL server 2022.

Tab. 3 – Hardwarové požadavky pro SQL server 2022 [30]

Komponenta	Požadavek
Pevný disk	Minimálně 6 GB volného místa na pevném disku.
Monitor	Vyžaduje monitor s rozlišením Super-VGA (800x600) nebo vyšším.
Internet	Vyžaduje přístup k internetu.
Paměť	Minimum: Express Edition: 512 MB, všechny ostatní edice: 1 GB. Doporučeno: Express Edition: 1 GB, všechny ostatní edice: Minimálně 4 GB a měly by být zvětšeny s rostoucí velikostí databáze, aby byl zajištěn optimální výkon.
Rychlost procesoru	Minimum: x64 Procesor: 1,4 GHz. Doporučeno: 2,0 GHz nebo rychlejší.
Typ procesoru	Procesor x64: AMD Opteron, AMD Athlon 64, Intel Xeon s podporou Intel EM64T, Intel Pentium IV s podporou EM64T.

4.2.3 Edice

- Enterprise – poskytuje komplexní funkce high-end datového centra s maximálním výkonem, neomezenou virtualizací a komplexní business inteligencí. Umožňuje vysokou dostupnost služeb pro kritické úlohy a koncových uživatelů k datům.
- Standard – umožňuje základní správu dat a databází, business inteligenci pro oddělení nebo malé organizace a provoz jejich aplikací, podporu běžných vývojových nástrojů pro místní a cloudové prostředí. Umožňuje efektivní správu databází s minimálními hardwarovými a softwarovými prostředky.
- Web Edition – edice s nízkými celkovými náklady na vlastnictví pro webové hostitele a webové VAP, která poskytuje škálovatelnost, cenovou dostupnost a možnosti správy pro malé i rozsáhlé webové služby.
- Developer – umožňuje vývojářům vyvíjet různé typy aplikací nad SQL Serverem. Zahrnuje všechny funkce edice Enterprise, ale je licencován jako vývojový a testovací systém, nikoli jako produkční server. SQL Server Developer je ideální volbou pro uživatele, kteří vytvářejí a testují aplikace.
- Express – základní, bezplatná edice, ideální pro výuku a tvorbu desktopových a malých serverových aplikací pracujících s daty. Volba pro nezávislé dodavatele softwaru, vývojáře a fanfy vytvářející klientské aplikace. Při potřebě pokročilejších databázových funkcí je možný upgrade na vyšší edice SQL Serveru. SQL Server Express LocalDB je „odlehčená“ verze Express, se všemi funkcemi, s rychlou instalací a bez nutnosti konfigurace.

[31]

4.3 MySQL

Je open-source databázový systém pro správu relačních databází. Název je kombinací „My“ (jméno dcery spoluzakladatele Michaela Wideniuse My) a „SQL“ (zkratka pro Structured Query Language). MySQL je bezplatný a open-source software za podmínek GNU General Public License a je také dostupný pod řadou proprietárních licencí. MySQL byl vlastněn a sponzorován švédskou společností MySQL AB, kterou koupila společnost Sun Microsystems (nyní Oracle Corporation). V roce 2010, kdy Oracle koupil Sun, Michael Widenius oddělil open-source projekt MySQL a vytvořil MariaDB.[32]

Tento databázový systém je světově nejoblíbenější. MySQL jako datovou základnu využívá mnoho nejpoužívanějších aplikací, vč. Facebooku, Twitteru, Netflixu, Uber, Airbnb, Shopify a Booking.com. Vzhledem k tomu, že MySQL je open-source, obsahuje řadu funkcí vyvinutých v úzké spolupráci s uživateli za více než 25 let.

MySQL se trvale řadí mezi nejoblíbenější vývojářské databáze pro jeho vysoký výkon, spolehlivost a snadné použití. Tento databázový systém podporuje následující vývojové jazyky a ovladače: PHP, Python, Java/JDBC, Node.js, Perl, Ruby, Go, Rust, C, C++, C#/.NET, ODBC.

Oblíbená je tzv. triáda, balíček Apache (webový server), PHP a MySQL, kterou používá mnoho začínajících programátorů pro jednoduchost instalace a bezplatné nekomerční použití.

Databázový systém MySQL je systém klient-server, který se skládá z více vláknového serveru SQL, který podporuje různé backendy, několika různých klientských programů a knihoven, nástrojů pro správu a široké škály rozhraní pro programování aplikací (API). MySQL také poskytuje vestavěnou více vláknovou knihovnu, kterou je možné propojit s uživatelskou aplikací a získat tak menší, rychlejší a snadněji spravovatelný samostatný produkt.[33]

4.3.1 Edice

MySQL je jako komerční produkt k dispozici ve 4 edicích:

- Classic Edition – ideální vestavěná databáze pro nezávislé výrobce softwaru, výrobce OEM a VAR vyvíjející aplikace náročné na čtení pomocí úložného modulu MyISAM. Vysoce výkonná a nulová administrační databáze.[34]
- Standard Edition – umožňuje poskytovat vysoce výkonné a škálovatelné aplikace OLTP při nižších nákladech, ale se stejnou silou, výkonem a spolehlivostí. Obsahuje InnoDB, což z ní činí plně integrovanou transakční databázi kompatibilní s ACID a dále replikace umožňující poskytovat vysoce výkonné a škálovatelné aplikace.[35]
- Enterprise Edition – obsahuje nejkomplexnější sadu pokročilých funkcí, nástrojů pro správu a technické podpory pro dosažení nejvyšší úrovně škálovatelnosti, zabezpečení, spolehlivosti a provozuschopnosti.[36]
- Cluster Carrier Grade Edition – distribuovaná databáze kombinující lineární škálovatelnost a vysokou dostupnost. Poskytuje přístup k paměti v reálném čase s transakční konzistencí napříč děleními a distribuovanými datovými sadami. Je určena pro kritické aplikace. MySQL Cluster má integrovanou replikaci mezi clustery napříč několika geografickými lokalitami. Ideální volba pro provoz na komoditním hardwaru a v globálně distribuované cloudové infrastruktuře.
[37]

MySQL 8.0 je poslední verze s instalačním programem MySQL. Od MySQL 8.1 je možné pro instalaci použít MSI nebo Zip archiv. MySQL 8.1 a vyšší obsahuje také balíček MySQL Configurator, nástroj pro konfiguraci MySQL Serveru.[38]

4.3.2 Verze

Tab. 4 uvádí stručný přehled vydaných verzí MySQL.

Tab. 4 – Přehled verzí MySQL [32]

Verze	Datum vydání	Poslední dílčí verze	Datum vydání poslední dílčí verze	Ukončení podpory
5.1	14. listopadu 2008	5.1.73	3. prosince 2013	prosinec 2013
5.5	3. prosince 2010	5.5.62	22. října 2018	prosinec 2018
5.6	5. února 2013	5.6.51	20. ledna 2021	únor 2021
5.7	21. října 2015	5.7.44	25. října 2023	říjen 2023
8.0	19. dubna 2018	8.0.36	16. ledna 2023	duben 2026
8.1	18. července 2023	8.1.26	18. července 2023	říjen 2023
8.2	25. října 2023	8.2.0	25. října 2023	leden 2024
8.3	16. ledna 2024	8.3.0	16. ledna 2024	duben 2024

Před nástupem MySQL AB společností Sun Microsystems existovala verze MySQL s číslem 6. Nový vlastník tuto verzi bohužel příliš nepodporoval a došlo k jejímu zániku. Verze 7 je řadu let používána pro produkt MySQL Cluster. S dalším vývojem MySQL 8 se vývojáři dohodli, že zapracované změny si „zaslouží“ další novou verzi 8.[39]

4.3.3 Systémové požadavky

Níže v Tab. 3 je uveden přehled základních hardwarových požadavků pro provoz databázového systému MySQL 8.2.

Tab. 5 - Hardwarové požadavky pro MySQL 8.2 [40]

Komponenta	Požadavek
Pevný disk	Schopný intenzivního zápisu databáze, doporučen RAID10 nebo RAID0+1. Požadované místo: Service manager <ul style="list-style-type: none"> • Windows – 800 MB • Linux – 1,3 GB • macOS – 1,2 GB Monitor agent: <ul style="list-style-type: none"> • Windows – 500 MB • Linux – 800 MB • macOS – 700 MB
Monitor	Vyžaduje monitor s rozlišením Super-VGA (800x600) nebo vyšším.
Internet	Vyžaduje přístup k internetu.
Paměť	Minimum: 2 GB. Doporučeno 8 GB.
Procesor	Minimum: 2 CPU. Doporučeno: 4 CPU a více.

5 Vybrané srovnávací atributy

PostgreSQL, Microsoft SQL Server (MSSQL) a MySQL jsou běžně používané relační databázové systémy. I když mají mnoho společných základních rysů, existují mezi nimi rozdíly. Asi nejvýznamnější rozdíl je, že PostgreSQL je open source, zatímco SQL server je vlastněn a licencován společností Microsoft a MySQL má omezení při komerčním použití. Další rozdíly mezi těmito systémy se týkají složitosti použití, syntaxe jazyka SQL a jejího dodržování, datových typů, dostupných funkcí, výkonu, zabezpečení a mnoho dalších. Srovnání by mělo ukázat úskalí přechodu z komerční databáze na databázi s otevřeným zdrojovým kódem, možné kompromisy a výhody (nebo nevýhody) těchto systémů. SQL Server je z historického hlediska oblíbený u organizací, které používají a spoléhají i na jiné produkty Microsoftu, ale PostgreSQL se postupně dostal na vrchol nejen díky výhodám open-source, ale také díky svým robustním funkcím a aktivní komunitě uživatelů.[41]

5.1 Licencování

PostgreSQL je open-source databáze vydaná pod licencí PostgreSQL License, Open Source Initiative Approved License. Použití PostgreSQL pro jakýkoli účel, včetně komerčního, je bezplatné. V rámci PostgreSQL Global Development Group je PostgreSQL k dispozici jako bezplatný software s otevřeným zdrojovým kódem na dobu neurčitou.

SQL Server je k dispozici prostřednictvím komerční licence a lze jej licencovat na úrovni jádra nebo na úrovni počtu přístupů klientů a serveru (CAL). SQL Server je nabízen ve dvou hlavních edicích, Enterprise Edition a Standard Edition, aby vyhověl požadavkům organizací a jednotlivců na výkon a cenu. Licenční náklady se pohybují od 3 586 USD za edici Standard do 13 748 USD za edici Enterprise (pro dvě jádra), pro model CAL jsou náklady 899 USD za server plus 209 USD na uživatele. Studentům a vývojářům je k dispozici bezplatná verze pro vytváření a testování.[41]

MySQL je open-source. Software může kdokoli používat a upravovat, a je možné si ho stáhnout z internetu a používat jej, aniž by bylo nutné za něj platit. Je možné také změnit jeho zdrojový kód tak, aby vyhovoval vlastním potřebám. Software MySQL používá GNU General Public License (GPL), která definuje, co se smí a nesmí dělat se softwarem v různých situacích. Pokud je potřeba vložit kód MySQL do komerční aplikace, je nutné si koupit komerčně licencovanou verzi od společnosti Oracle.[33]

5.2 Snadnost použití

PostgreSQL je pokročilý objektově-relační systém pro správu databází, který kromě vlastního procedurálního jazyka PL/pgSQL využívá i strukturovaný dotazovací jazyk (SQL). PostgreSQL je snadno použitelný s celou řadou databázových funkcí RDBMS a možností pro práci s daty. Lze jej snadno nainstalovat do prostředí provozovaných na Linuxu, ale i do ostatních OS.

SQL Server je systém správy relačních databází (RDBMS) vyvinutý a provozovaný společností Microsoft. Používá variantu strukturovaného dotazovacího jazyka (SQL) nazvanou T-SQL (Transact-SQL). Může běžet na operačních systémech Linux s podporou Kubernetes, ale hlavně běží na operačních systémech Windows. Má kompatibilitu s .NET Framework.[41]

MySQL je možné během několika minut nainstalovat a nakonfigurovat. Správa databáze je velmi snadná. Stejně jako PostgreSQL lze nainstalovat do všech OS.[42]

Pokud jde o podporu programovacích jazyků, existuje poměrně velký rozdíl mezi PostgreSQL a SQL Serverem. PostgreSQL podporuje Python, PHP, Perl, Tcl, .NET, C/C++, Delphi, Java, JavaScript (Node.js) a další. SQL Server je omezenější, nabízí podporu pro Javu, JavaScript (Node.js), C#, C++,

PHP, Python a Ruby. MySQL podporuje C/C++, Delphi, Erlang, Go, Java, Lisp, Node.js, Perl, PHP a R.[42][43]

5.3 Syntaxe

PostgreSQL i SQL Server používají standardní SQL dotazovací jazyk, ale mají také implementovanou vlastní verzi SQL jazyka, tzv. dialekt SQL. SQL Server používá T-SQL, který poskytuje veškerou funkcionalitu SQL a přidává několik proprietárních programovacích rozšíření. V PostgreSQL je možné použít SQL s vlastním procedurálním jazykem PL/pgSQL, který umožňuje vytvářet funkce, spouštět procedury a přidávat vlastní ovládací prvky struktur do SQL.[43]

MySQL používá standardní SQL, ale nepodporuje poddotazy, jako je „LIMIT“ nebo „ALL“. Nepodporuje ani klauzule SQL jako „INTERSEC“ nebo „OUTER JOIN“. MySQL tak není plně kompatibilní s SQL jako oba další systémy, které podporují všechny výše uvedené dílčí dotazy.[42]

Syntaxe všech tří systémů je velmi podobná s několika drobnými rozdíly. V Tab. 6 je několik příkladů nejzákladnějších rozdílů syntaxe mezi PostgreSQL a SQL Server.[43]

Tab. 6 – Rozdíly syntaxe SQL mezi PostgreSQL a SQL Server [41]

	PostgreSQL	SQL Server
příkaz SELECT	SELECT col1, col2	SELECT [col1], [col2]
alias pro sloupce a tabulky	SELECT AVG(col1) AS avg1	SELECT AVG(col1) = avg1
práce s daty	CURRENT_DATE() CURRENT_TIME() EXTRACT()	GETDATE() DATEPART()

5.4 Datové typy

V Tab. 7 jsou uvedeny datové typy používané v PostgreSQL, SQL Server a MySQL.

Tab. 7 – Používané datové typy [41][44]

Datový typ	PostgreSQL	SQL Server	MySQL
64bitové celé číslo	BIGINT	BIGINT	BIGINT(n)
byte řetězec s pevnou délkou	BYTEA	BINARY(n)	BINARY(n)
1, 0 nebo NULL	BOOLEAN	BIT	BOOL (BOOLEAN)
řetězec znaků s pevnou délkou, 1 ≤ n ≤ 8000	CHAR(n)	CHAR(n)	CHAR(n)
řetězec znaků s proměnnou délkou, 1 ≤ n ≤ 8000	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)
řetězec znaků s proměnnou délkou, ≤ 2 GB	TEXT	VARCHAR(max)	VARCHAR(n)
byte řetězec s proměnnou délkou, 1 ≤ n ≤ 8000	BYTEA	VARBINARY(n)	(x)BLOB
byte řetězec s proměnnou délkou, ≤ 2 GB	BYTEA	VARBINARY(max)	(x)BLOB
řetězec Unicode s proměnnou délkou	VARCHAR(n)	NVARCHAR(n)	TEXT
data Unicode s proměnnou délkou, ≤ 2 GB	TEXT	NVARCHAR(max)	TEXT
data s proměnnou délkou, ≤ 2 GB	TEXT	TEXT	TEXT

data Unicode s proměnnou délkou, <= 2 GB	TEXT	NTEXT	TEXT
číslo s plovoucí desetinnou čárkou s dvojitou přesností	DOUBLE PRECISION	DOUBLE PRECISION	DOUBLE PRECISION
číslo s plovoucí desetinnou čárkou	DOUBLE PRECISION	FLOAT(p)	DOUBLE
32bitové celé číslo	INTEGER	INTEGER	INT (INTEGER)
datum obsahující rok, měsíc a den	DATE	DATE	DATE
datum a čas se zlomky sekund	TIMESTAMP(p)	DATETIME, DATETIME2(p)	DATETIME2
datum a čas s časovým pásmem	TIMESTAMP(p) WITH TIME ZONE	DATETIMEOFFSET (p)	TIMESTAMP(p)
datum a čas	TIMESTAMP(0)	SMALLDATETIME	TIMESTAMP(0)
celé číslo bez znaménka, 0 to 255 (8 bitů)	SMALLINT	TINYINT	TINYINT(n)
UUID – unikátní identifikátor (16 bajtů)	CHAR(16)	UNIQUEIDENTIFIER	BINARY, VARBINARY, VARCHAR
automaticky aktualizovaná binární data	BYTEA	ROWVERSION	NE (lze obejít programově)
částka v měně (32 bitů)	MONEY	SMALLMONEY	DECIMAL(n, d)
binární data s proměnnou délkou, <= 2 GB	BYTEA	IMAGE	BLOB
geometrické typy	POINT, LINE, LSEG, BOX, PATH, POLYGON, CIRCLE	GEOMETRY	POINT, LINestring, POLYGON, GEOMETRYCOLLECTION

5.5 Geografická data

PostgreSQL nemá nativní datový typ pro geografická data. Open-source doplněk PostGIS nabízí podporu pro geografické objekty. SQL Server má datový typ geografických dat pro ukládání geografických prostorových dat. MySQL nemá nativní datový typ pro geografická data, lze využít datový typ DECIMAL, pro zeměpisnou šířku DECIMAL(10, 8) a pro zeměpisnou délku DECIMAL(11, 8).[41][45]

5.6 Rozlišování malých a velkých písmen (case sensitive)

PostgreSQL rozlišuje malá a velká písmena pro vyhodnocování řetězců. Funkce LOWER() umožňuje uživatelům převádět řetězce na všechna malá písmena pro účely vyhodnocení (existuje také podobná funkce UPPER()). Ve výchozím nastavení PostgreSQL převádí názvy tabulek a sloupců na malá písmena, pokud tyto názvy nejsou umístěny v uvozovkách. Modul CITEXT poskytuje řetězec dat typu CITEXT bez rozlišení velkých a malých písmen pro porovnávání hodnot.

SQL Server ve výchozím nastavení nerozlišuje malá a velká písmena. Rozlišování malých a velkých písmen lze změnit úpravou nastavení řazení serveru SQL Server. Nastavení řazení pro rozlišování malých a velkých písmen lze nastavit na úrovni databáze nebo sloupce.[41]

MySQL u nebinárních řetězců (CHAR, VARCHAR, TEXT) používá při hledání řetězců řazení znaků porovnávaných operandů, velká a malá písmena se nerozlišují. U binárních řetězců (BINARY, VARBINARY, BLOB) se používá porovnání číselné hodnoty bajtů v operandech, to znamená, že u abecedních znaků se při porovnání rozlišují velká a malá písmena.[46]

5.7 Indexy

PostgreSQL nabízí řadu možností pro typy indexů, včetně B-stromu, hash, Generalized Search Tree (GiST), Space Partitioned GiST, Generalized Inverted Index (GIN) a Block Range Index (BRIN). Kromě toho podporuje výrazové indexy (indexy vytvořené výrazem nebo funkcí nad hodnotou sloupce) a částečné indexy (indexy části tabulky).

SQL Server nabízí tzv. seskupené (clustered) a neseskupené (non-clustered) indexy. Seskupené indexy třídí datové řádky v tabulce nebo view na základě hodnot klíče (sloupce v definici indexu). Tabulka může mít pouze jeden seskupený index. Neseskupené indexy jsou uloženy odděleně od dat tabulky a každý záznam hodnoty klíče má ukazatel, kde se data v tabulce nacházejí. SQL Server indexy vytváří automaticky, pokud jsou ve sloupcích tabulky definovány omezení Primary Key a Unique Key. Omezení Unique Key vytvoří neseskupený index, zatímco Primary Key vytvoří seskupený index, pokud již takový neexistuje.[41]

MySQL většinu indexů (PRIMARY KEY, UNIQUE, INDEX a FULLTEXT) ukládá v B-stromech. Výjimkou jsou indexy na typech prostorových dat, které používají R-stromy. MEMORY tabulky také podporují hash indexy a InnoDB používá invertované seznamy pro FULLTEXT indexy.[47]

5.8 Replikace

PostgreSQL má primárně-sekundární replikaci. Replikace může být synchronní nebo asynchronní. Asynchronní replikace se provádí pomocí Write-Ahead logů (WAL) pro sdílení změn s replikačními uzly. Tato replikace dat umožňuje odběratelským serverům, aby byly aktualizovány okamžitě poté, co jsou data na publikačním serveru zpracována. Logická replikace se řídí modelem publikování a odběru – tato metoda replikace se nazývá logická, protože změny jsou založeny na identitě replikace dat (například primární klíč) než na jejich fyzickém umístění. Fyzická replikace naopak pracuje se soubory a adresáři bez ohledu na obsah v těchto fyzických umístěních. PostgreSQL nativně nenabízí multi-master replikaci (replikaci mezi publikačními servery), ale některé nástroje třetích stran toto multi-master replikační řešení umožňují.

Replikace SQL Serveru nabízí tři typy replikace:

- transakční replikace pro prostředí server-server, kde jsou změny doručovány publikačním serverem k odběratelskému, jakmile k nim dojde.
- slučovací replikace pro prostředí server-klient nebo v situacích, kdy by mohlo dojít ke konfliktům a kdy lze data měnit a sledovat na publikačním i odběratelském serveru a později je synchronizovat.
- snapshot replikace, když jsou data aktualizována zřídka nebo není třeba je měnit postupně a kdy jsou data duplikována přesně tak, jak se v konkrétním okamžiku objevují.

Replikace v SQL Serveru může být synchronní nebo asynchronní. Verze Enterprise nabízí replikaci peer-to-peer jako alternativní řešení replikace s více publikačními servery.[41]

MySQL replikace podporuje různé typy synchronizací. Původní typ je jednosměrná asynchronní replikace, ve které jeden server funguje jako zdroj, zatímco jeden nebo více dalších serverů funguje jako repliky. Dále synchronní replikaci, která je charakteristická pro NDB Cluster. V MySQL 8.0 je kromě vestavěné asynchronní replikace podporována i semisynchronní replikace.

U semisynchronní replikace je potvrzení provedené na zdrojových blocích před návratem k relaci, která provedla transakci, pokud alespoň jedna replika nepotvrdí, že přijala a zaprotokolovala události transakce. Tato nejnovější verze MySQL také podporuje zpožděnou replikaci, takže replika záměrně zaostává za zdrojem alespoň o určitou dobu.[47]

5.9 Clustering

PostgreSQL umožňuje clusteru serverů, ale nativně nepodporuje multi-master nebo active/active clusteru. Nástroje jako RepMgr (replication manager) umožňují snadnou údržbu clusterů PostgreSQL.

SQL Server nabízí službu Windows Server Failover Clustering, kterou lze nakonfigurovat pro active/passive i active/active uzly. Edice SQL Serveru Standard podporuje pouze dva uzly pro cluster, přidání dalších uzlů vyžaduje upgrade na edici Enterprise.[41]

MySQL umožňuje klastrování pomocí NDB Cluster (edice NDB Cluster CGE), což je distribuovaná databáze kombinující lineární škálovatelnost a vysokou dostupnost. Poskytuje přístup v reálném čase v paměti s transakční konzistencí napříč rozdělenými a distribuovanými datovými sadami. NDB Cluster má vestavěnou replikaci mezi clusteru napříč více geografickými lokalitami.[48]

5.10 Vysoká dostupnost

PostgreSQL nabízí řadu řešení k zajištění vysoké dostupnosti pro uživatele, včetně převzetí služeb při selhání sdíleného disku, odesílání protokolů s předběžným zápisem, dělení dat a několika metod replikace. Nástroje jako Postgres Failover Manager poskytují automatické převzetí služeb po havárii a zajištění vysoké dostupnosti monitorováním a identifikací selhání databáze.

SQL Server obsahuje ve svých edicích řadu nástrojů pro vysokou dostupnost. Jsou to např. replikace, odesílání protokolů a clusteru s podporou převzetí služeb při selhání. Funkce skupiny dostupnosti Always On, která je součástí edice Enterprise, nabízí řešení vysoké dostupnosti a zotavení po havárii pro klíčové databáze.[41]

MySQL InnoDB Cluster (edice Enterprise) poskytuje kompletní řešení vysoké dostupnosti. Každý server v klastru InnoDB replikuje data do všech členů klastru a zároveň poskytuje odolnost proti chybám, automatické převzetí služeb při selhání a pružnost. MySQL InnoDB Cluster poskytuje vestavěnou správu členství ve skupinách, záruky konzistence dat, detekci selhání uzlů a převzetí služeb při selhání databáze bez nutnosti ručního zásahu.[49]

5.11 Views

PostgreSQL views (česky pohledy, virtuální tabulky, které samy neukládají data) podporuje. Jsou podporovány aktualizovatelné pohledy, ale aktualizace neprobíhají automaticky, pokud nespĺňují následující podmínky:

- dotaz takového pohledu musí mít přesně jeden oddíl v klauzuli FROM a může to být tabulka nebo jiný aktualizovatelný pohled.
- výběrový seznam nesmí obsahovat žádnou funkci okna, agregační funkce ani žádnou funkci vracející sadu dat.
- dotaz nesmí obsahovat jednu z následujících klauzulí na nejvyšší úrovni – HAVING, LIMIT, DISTINCT, WITH, INTERSECT, EXCEPT, OFFSET AND LIMIT.

Pohledy vytvořené jednoduchými dotazy lze tedy aktualizovat. Ty vytvořené pomocí složitých dotazů nelze, ale komplexní pohledy lze aktualizovat pomocí pravidel. Podporovány jsou také tzv.

materializované pohledy. Data v materializovaných pohledech lze aktualizovat pomocí příkazu REFRESH MATERIALIZED VIEW.

Views v SQL Server se používají pro účely zabezpečení a k omezení přístupu uživatelů k datům. Jsou podporovány pohledy definované uživatelem i systémem. Pohledy lze automaticky aktualizovat pomocí spouštěčů. Data v pohledu lze aktualizovat, když jsou provedeny úpravy sloupce z jedné základní tabulky a jsou přímo odkazovány. Materializovaná zobrazení jsou na SQL Server známá jako indexovaná zobrazení; na rozdíl od materializovaných pohledů v jiných relačních databázích jsou indexované pohledy synchronizovány s podkladovými daty a jsou tak aktualizovány automaticky.[41]

V MySQL je view virtuální tabulka vytvořená dotazem spojením jedné nebo více tabulek. Funguje podobně jako základní tabulka, ale neobsahuje žádná vlastní data. Pohled a tabulka mají jeden hlavní rozdíl v tom, že pohledy jsou definice postavené na jiných tabulkách (nebo pohledech). Pokud dojde k jakýmkoli změnám v podkladové tabulce, stejné změny se projeví také v zobrazení.[50]

5.12 Triggery

PostgreSQL má pokročilé triggery (česky spouštěče). Podporované spouštěcí události jsou AFTER, BEFORE a INSTEAD OF a lze je použít pro události INSERT, UPDATE a DELETE. Funkce lze použít ke spouštění komplexních SQL dotazů, když je trigger aktivován a PostgreSQL může toto provádět dynamicky.

SQL Server nabízí spouštěče pro různé typy databázových událostí:

- DML triggery – pro konkrétní událost jazyka manipulace s daty (DML), jako je vkládání, aktualizace nebo mazání záznamů. Tyto spouštěče spouštějí události bez ohledu na počet ovlivněných řádků.
- DDL triggery – pro události jazyka definice dat (DDL), jako jsou příkazy CREATE, DROP nebo ALTER. Tyto triggery jsou užitečné pro prevenci nebo auditování změn schématu databáze.
- Logon triggery – pro události přihlášení, například při vytvoření uživatelské relace. Tyto spouštěče se spouštějí po úspěšné autentizaci a před navázáním uživatelské relace. Jsou užitečné pro auditování a řízení přihlašovací aktivity.[41]

MySQL aktivuje triggery pouze pro změny provedené v tabulkách příkazy SQL. To zahrnuje změny základních tabulek, které jsou základem aktualizovatelných pohledů. Spouštěče se neaktivují pro změny v tabulkách provedené rozhraními API, která nepřenášejí příkazy SQL na server MySQL. To znamená, že spouštěče nejsou aktivovány aktualizacemi provedenými pomocí NDB API. Spouštěče nejsou aktivovány změnami v tabulkách INFORMATION_SCHEMA nebo PERFORMANCE_SCHEMA. Tyto tabulky jsou ve skutečnosti pohledy a spouštěče nejsou v pohledech povoleny.[51]

5.13 Fulltextové vyhledávání

PostgreSQL nabízí pokročilé funkce pro fulltextové vyhledávání. Pro rychlejší vyhledávání využívá fulltextové indexování a slovníky. Předzpracované textové dokumenty se ukládají jako datový typ TSVECTOR a zpracované dotazy se ukládají jako typ TSQUERY. Předzpracování analyzuje textové dokumenty do jazykových jednotek známých jako lexémy, což vám umožňuje najít varianty slova bez rozlišení velkých a malých písmen.

SQL Server nabízí fulltextové vyhledávání jako volitelnou součást. Dotazy jsou spouštěny proti fulltextovému indexu a vyhledávání může být založeno na pravidlech určitého jazyka. Vyhledávání se provádí ve sloupcích nebo textových typech dat (včetně znaků CHAR, VARCHAR, NCHAR, NVARCHAR, TEXTU, NTEXT, IMAGE, XML nebo VARBINARY(MAX) a FILESTREAM) pomocí T-SQL příkazu CONTAINS pro nalezení slov a frází a příkazu FREETEXT k nalezení významu. Soubory Tezauru (normalizovaný strukturovaný slovník) lze použít k nalezení synonym hledaných výrazů. Fulltextové vyhledávání na serveru SQL Server není case sensitive (nerozlišují se velká a malá písmena).[41]

MySQL má podporu pro fulltextové indexování a vyhledávání. Fulltextový index v MySQL je index typu FULLTEXT. Fulltextové indexy lze použít pouze s tabulkami InnoDB nebo MyISAM a lze je vytvořit pouze pro sloupce typu CHAR, VARCHAR nebo TEXT. MySQL poskytuje vestavěný fulltextový analyzátor ngramů (souvislá posloupnost n znaků z dané posloupnosti textu), který podporuje čínštinu, japonštinu a korejštinu (CJK), a instalovatelný plugin pro fulltextový parser MeCab pro japonštinu.[52]

5.14 Celá čísla

V PostgreSQL existují tři druhy celých čísel, jejich přehled, rozsah a vyžadovanou velikost paměťového prostoru poskytuje Tab. 8.

Tab. 8 – Celá čísla v PostgreSQL [41]

Datový typ	Rozsah	Pam. prostor
BIGINT	-2^{63} (-9 223 372 036 854 775 808) až $2^{63}-1$ (9 223 372 036 854 775 807)	8 bajtů
INT	-2^{31} (-2 147 483 648) až $2^{31}-1$ (2 147 483 647)	4 bajty
SMALLINT	-2^{15} (-32 768) až $2^{15}-1$ (32 767)	2 bajty

SQL Server podporuje standardní SQL celočíselné typy BIGINT, INT, SMALLINT a TINYINT. Rozsah a velikost paměťového prostoru typu jsou uvedeny v Tab. 9.

Tab. 9 – Celá čísla v SQL Server [41]

Datový typ	Rozsah	Pam. prostor
BIGINT	-2^{63} (-9 223 372 036 854 775 808) až $2^{63}-1$ (9 223 372 036 854 775 807)	8 bajtů
INT	-2^{31} (-2 147 483 648) až $2^{31}-1$ (2 147 483 647)	4 bajty
SMALLINT	-2^{15} (-32 768) až $2^{15}-1$ (32 767)	2 bajty
TINYINT	0 až 255	1 bajt

MySQL podporuje standardní celočíselné typy SQL INTEGER (nebo INT) a SMALLINT. Jako rozšíření standardu podporuje MySQL také celočíselné typy TINYINT, MEDIUMINT a BIGINT. Tab. 10 poskytuje přehled o požadovaném paměťovém prostoru a rozsah pro každý typ celého čísla.

Tab. 10 – Celá čísla v MySQL [53]

Datový typ	Rozsah	Pam. prostor
BIGINT	-2^{63} (-9 223 372 036 854 775 808) až $2^{63}-1$ (9 223 372 036 854 775 807)	8 bajtů
INT	-2^{31} (-2 147 483 648) až $2^{31}-1$ (2 147 483 647)	4 bajty
MEDIUM	-2^{23} (-8 388 608) až $2^{23}-1$ (8 388 607)	3 bajty
SMALLINT	-2^{15} (-32 768) až $2^{15}-1$ (32 767)	2 bajty
TINYINT	0 až 255	1 bajt

5.15 Booleovské typy

Booleovský datový typ PostgreSQL může mít 3 stavy: TRUE, FALSE a NULL.

V SQL serveru se používá datový typ BIT k reprezentaci pravdivých/nepravdivých booleovských dat. Hodnota pole BIT je 1, 0 nebo NULL.[41]

MySQL nemá datový typ BOOLEAN (nebo BOOL) a převádí booleovské hodnoty na celočíselné datové typy (TINYINT). Pokud je vytvořena tabulka s datovým typem BOOLEAN, MySQL zobrazí data jako 0, pokud je hodnota FALSE, nebo jako 1, pokud je hodnota TRUE.[54]

5.16 NoSQL podpora

PostgreSQL, stejně jako mnoho dalších relačních databází, přidal podporu pro data JSON – nejběžnější formát pro polostrukturovaná data uložená v systémech NoSQL. Ale protože SQL je jediný způsob interakce s databází PostgreSQL, nedá se toto považovat za NoSQL.

SQL Server má nativní funkce JSON, které umožňují analyzovat dokumenty JSON pomocí standardního jazyka SQL. Dokumenty JSON je možné ukládat na SQL Server a dotazovat se na tyto data stejně jako v databázi NoSQL. Protože SQL Server je SQL databáze, i zde se toto nedá považovat za NoSQL.[41]

MySQL NDB Cluster umožňuje uživatelům kombinovat relační a NoSQL technologii. MySQL NDB Cluster představuje několik rozhraní k databázi, což umožňuje maximální agilitu vývojářů a zároveň poskytuje možnost zcela obejít vrstvu SQL pro nativní a rychlý přístup k tabulkám. Každé z SQL a NoSQL API lze používat současně, v rámci stejné datové sady. Server MySQL poskytuje clusteru standardní rozhraní SQL, které umožňuje spouštění složitých relačních dotazů a poskytuje připojení ke všem standardním konektorům MySQL včetně PHP, Perl, Python, Django, Ruby, Ruby on Rails, JDBC (pro další připojení k ORM včetně EclipseLink nebo Hibernate) .NET, ODBC atd.[55]

5.17 Bezpečnost

PostgreSQL podporuje SSL připojení (Secure Sockets Layer – zabezpečení komunikace přes internet) pro šifrování komunikace klient-server. SSL lze povolit nastavením příslušného parametru v konfiguračním souboru postgresql.conf. Pro další funkce, jako audit, správa hesel a možnosti redakce dat je možné použít nástroje třetích stran jako EDB Postgres Advanced Server.

SQL Server nabízí řadu funkcí proti bezpečnostním hrozbám, protože každá aplikace má specifické bezpečnostní potřeby. V rámci zabezpečení SQL Server spravuje přístup k zabezpečeným entitám prostřednictvím ověřování a autorizace. SQL Server má také podporu pro hierarchii pomocí SSL šifrování a podporuje TLS (Transport Layer Security – zabezpečení transportní vrstvy) pro šifrování síťového provozu.[41]

MySQL v zabezpečení dat zahrnuje ochranu v souladu s průmyslovými a vládními předpisy, včetně obecného nařízení Evropské unie o ochraně osobních údajů, standardu pro bezpečnost dat v odvětví platebních karet, zákona o přenositelnosti a odpovědnosti zdravotního pojištění a bezpečnostních technických pokynů pro implementaci Agentury pro obranné informační systémy. MySQL Enterprise Edition poskytuje pokročilé funkce zabezpečení, včetně ověřování a autorizace, transparentního šifrování dat, auditování, maskování dat a databázového firewallu.[33]

5.18 Nástroje pro správu – administrace pomocí GUI

PostgreSQL lze v prostředí OS Windows administrovat pomocí tzv. GUI nástrojů jako pgAdmin, OmniDB, DBeaver a Postgres Enterprise Manager. Mezi další GUI nástroje používané pro monitorování funkčnosti a výkonu patří např. Nagios, Zabbix, Cacti a EDB Postgres. SQLLECTRON je

multiplatformní bezplatná open-source varianta, která je kompatibilní s řadou SQL databází včetně SQL Serveru.

SQL Server lze v prostředí OS Windows spravovat prostřednictvím GUI pomocí SQL Server Management Studia (SSMS), které je zdarma a součástí každé instalace. Lze je nainstalovat i samostatně (např. na klientský počítač) a administrovat vzdálené servery. SQL Operations Studio je bezplatné, open-source, multiplatformní GUI vč. Mac.[41]

MySQL lze v prostředí OS Windows pomocí GUI ovládat např. přes MySQL Workbench, phpMyAdmin, Navicat, Dbeave nebo TablePlus. Většina z těchto nástrojů jsou bezplatné a multiplatformové (je možné jimi ovládat i jiné databázové systémy).[56]

5.19 Souběžnost

PostgreSQL má dobře vyvinuté řízení souběžnosti více verzí MVCC (MultiVersion Concurrency Control) pro zpracování více procedur najednou. MVCC poskytuje snímky databázových informací, aby se zabránilo zobrazení nekonzistencí způsobených simultánními transakcemi nebo zamykáním dat, ke kterým dochází v jiných databázových systémech. K zaručení izolace transakcí používá Serializable Snapshot Isolation (SSI).

SQL Server má méně plně vyvinutý systém řízení souběžnosti s více verzemi a ve výchozím nastavení „spoléhá“ na zamykání dat, aby se zabránilo chybám ze simultánních transakcí. Nabízí optimistický rys souběžnosti, který předpokládá, že se takové konflikty objevují jen zřídka; místo zamykání řádku je porovnán s verzí uloženou v mezipaměti, aby se zjistilo, zda došlo k nějaké změně.[41]

MySQL používá vlákna operačního systému ke zpracování požadavků z uživatelských transakcí. V situacích, kdy je užitečné minimalizovat přepínání kontextu mezi vlákny, může MySQL použít řadu technik k omezení počtu souběžně spouštěných vláken operačního systému (a tím i počtu požadavků, které jsou zpracovány v jednom okamžiku). Když MySQL obdrží nový požadavek z uživatelské relace, a pokud je počet souběžně prováděných vláken na předem definovaném limitu, nový požadavek na krátkou dobu „uspí“, než se pokusí znovu jej zpracovat. Požadavek, který nelze po uspání přeplánovat, je zařazen do fronty first-in/first-out a nakonec je zpracován. Vlákna čekající na uzamčení se nezapočítávají do počtu souběžně běžících vláken.[57]

5.20 Výkon

PostgreSQL nabízí rychlost a výkon napříč datovými sadami všech velikostí a pravidelně překonává ostatní databáze v rychlosti zpracování online transakcí (OLTP) i online analytického zpracování (OLAP). Nabízí řízení souběžnosti více verzí (MVCC), které umožňuje zpracovávat více transakcí současně s menším počtem kolizí než SQL Server. PostgreSQL nabízí mnoho nástrojů, které uživatelům umožňují sledovat výkon databáze, a parametrů, které umožňují optimalizovat výkon databáze.

SQL Server je rychlý pro analytické i transakční zpracování dat. Bohužel společnost Microsoft zakazuje publikování srovnávacích testů bez jejího předchozího písemného souhlasu, a tak přímá srovnání s jinými databázovými systémy jsou vzácná. Mezi hlavní funkce SQL Serveru pro optimalizaci výkonu a rychlosti patří jeho In-Memory OLTP, využívající datové tabulky v paměti, které jsou mnohem rychlejší než přímý zápis na disk. Edice SQL Serveru Standard obsahuje určitá omezení výkonu paměti, indexování a dalších funkcí, a v případě potřeby navýšení rychlosti a výkonu, je nutný upgrade na verzi Enterprise.[41]

MySQL je podle standardních benchmarků (včetně TPC-H, TPC-DS a CH-benCHmark) rychlejší než jiné databázové služby. Bohužel toto se v praktické části (kdy bylo ponecháno defaultní nastavení) nepotvrdilo.[33]

5.21 Oblíbenost

PostgreSQL je nejpokročilejší open-source databáze na světě. Je využíván podniky po celém světě pro kritické úlohy. Zejména komunita PostgreSQL a i několik společností, jako je EnterpriseDB nebo 2ndQuadrant, poskytují podporu, aby přechod podniku na PostgreSQL a používání tohoto databázového systému nebylo problematické.

SQL Server je oblíbený u podniků, které spoléhají a využívají produkty společnosti Microsoft. V posledních dvou desetiletích byl zaznamenán nárůst podílu na trhu (i z důvodu podpory a propagace přes Windows servery), ale jak v posledních letech stále více podniků přechází na open-source software, popularita SQL serveru klesá.[41]

MySQL je rychlý, spolehlivý, škálovatelný a snadno použitelný. Původně byl tento databázový systém vyvinut pro rychlou práci s velkými databázemi a již mnoho let se používá ve vysoce náročných produkčních prostředích. Přestože se MySQL neustále vyvíjí, nabízí bohatou a užitečnou sadu funkcí. Díky konektivě, rychlosti a zabezpečení je databázový systém vhodný pro přístup k databázím na internetu.[33]

5.22 Souhrn

V Tab. 11 je uveden souhrn nejdůležitějších srovnávacích kritérií.

Tab. 11 – Přehled srovnávacích kritérií

Kritérium	PostgreSQL	SQL Server	MySQL
Podporované typy databází	Objektově-relační	Relační	Relační
Licence	Open-source (s otevřeným kódem)	Komerční produkt (placené licence založené na jádru nebo uživatelích, bezplatná licence edice Express s omezenými funkcemi a velikostí databáze)	Open-source (pro komerční použití je nutné zakoupit placenou licenci)
Podporované platformy	Windows, Linux, macOS, BSD a Solaris. Je možné ho také nasadit na kontejnery Dockeru nebo Kubernetes.	Microsoft nebo Linux	Windows, Linux, macOS, FreeBSD a Solaris
Typ SQL	Standardní SQL	Transact-SQL nebo T-SQL (standardní SQL + proprietární programovací rozšíření)	Standardní SQL (nepodporuje některé poddotazy, jako je „LIMIT“ nebo „ALL“ a klauzule jako „INTERSEC“ nebo „OUTER JOIN“)
Podpora geografických dat	Ne	Ano	Ne (Ize využít datové typ DECIMAL)
Case sensitive	Ano	Pouze jako volitelná součást	Ne u nebinárních řetězců, Ano u binárních řetězců
Indexy	Ano	Ano	Ano
Replikace	Ano	Ano	Ano
Clustering	Ano	Ano	Ano
Views	Ano	Ano	Ano
Trigery	Ano	Ano	Ano
Fulltextové vyhledávání	Ano	Ano	Ano
NoSQL podpora	Ano, pomocí jazyka SQL	Ano, pomocí nativních funkcí	Ano
Zabezpečení	SSL + šifrování dat	SSL + šifrování dat na úrovni sloupců	Šifrování dat
Souběžnost	Ano	Ano	Ano

6 Zátěžové testy

Charakteristiky databázových operací v kontextu pracovní zátěže ovlivňují výkon databázových systémů. Tento problém také komplikuje zpracování transakcí. Obecně platí, že operace **INSERT**, **SELECT**, **UPDATE** a **DELETE** mají každá jiné pracovní zatížení, které významně ovlivňuje latenci a výsledky propustnosti.

Pracovní zátěž je posloupnost operací, které definují přístupové vzory a variace operací (zpracovávaných podle sekvence nebo času) v databázi. Pracovní zátěže jsou založeny buď na datech založených na trasování/reálném světě, nebo na syntetických datech, které systémy vytvářejí pomocí vzorů zatížení a omezení z reálného světa nakonfigurovaných pro daný proces. Existují dva primární typy benchmarků: syntetické a benchmarky „reálného světa“.[22]

6.1 Syntetické benchmarky

Obvykle simulují pracovní zatížení, jako je pracovní zatížení OLTP nebo standardní benchmark jako v TPC-C nebo TPC-H. Simulují pracovní zátěž v reálném světě tím, že se pokoušejí postupovat podle stejných vzorů, aby určily, jak stávající konfigurace databáze a hardware spolupracují při daném typu zátěže. Syntetické benchmarky mohou běžet kdekoli, nezávisle na návrhu schématu, nemohou však nahradit reálné podmínky.[22]

6.2 Srovnávací testy v reálném světě

Jsou založené na trasování, používají sadu dat a dotazy související s aplikací. Nevyžadují nutně úplnou datovou sadu databázového benchmarku a kompletní mix dotazů. Cílem je „vidět“ a pochopit přesné interakce mezi aplikací, konfigurací databáze a hardware, ať už s konkrétním aspektem nebo obecně.[22]

6.3 Důvody pro srovnávání databáze

Existuje několik důvodů pro benchmarking databáze:

- výběr správné databáze – existuje mnoho databází, z nichž mnohé se specializují na konkrétní případy použití. Pro každý z nich obvykle existuje několik vhodných databází. Databázový benchmarking nabízí srovnatelné metriky pro každou z nich, a slouží jako podklad pro kvantitativní a objektivní rozhodnutí o nejlepší možnosti, pracovní zátěži a dalších parametřích.
- výběr cloudového úložiště – výběr poskytovatele cloudu může ovlivnit všechny funkční vlastnosti, jako je dostupnost, výkon a škálovatelnost. Výsledky databázových benchmarků se totiž mohou mezi poskytovateli cloudu výrazně lišit. Databáze nebo pracovní zátěž může u jednoho poskytovatele vynikat, ale u jiného selhat. Zatímco úplná predikce je nemožná, srovnávání databáze může identifikovat korelace a ideální shody. Srovnávání databáze nabízí základ pro rozhodnutí založené na výsledcích testů o tom, který poskytovatel cloudu nejlépe vyhovuje pracovní zátěži jako interní úložiště, typ virtuálního počítače, poměr výkonu a nákladů a další.
- sledování změn výkonu – rutinní srovnávání databází před i po aktualizacích verzí odhalí rozdíly ve výkonu a umožní vývojovým týmům je proaktivně řešit.
- plánování kapacity – jak požadavky na pracovní zátěž kolísají, možnosti a limity systémů jsou často neznámé. Zátěžové testování identifikuje limity aktuálního nastavení a plánování kapacity umožňuje lepší bezpečnost a rychlejší reakční doby.

- ladění systému – pravidelné ladění databázových nastavení za účelem obcházení úzkých míst, snížení existujících problémů nebo zmírnění předpokládaných problémů je náchylné k chybám, časově náročné a nepříjemné, ale mnohem méně, pokud správce má k dispozici optimální nastavení z benchmarkingu databáze.

[22]

6.4 TPC-A

Srovnávací test TPC-A, vydaný v listopadu 1989 sdružením TPC (www.tpc.org), měří výkon v databázových prostředích náročných na aktualizaci, typických pro aplikace se zpracováním online transakcí (OLTP – OnLine Transfer Protocol). Taková prostředí se vyznačují:

- více online terminálovými relacemi
- významnými diskovými vstupy/výstupy
- středními časy zpracování systémem a aplikací
- transakční integritou

Tento benchmark používá k načtení testovaného systému jedinou, jednoduchou transakci náročnou na aktualizaci. Pracovní zátěž je tedy zaměřena tak, aby odrážela aplikaci OLTP, ale neodráží celý rozsah požadavků OLTP, které jsou typicky charakterizovány více typy transakcí s různou složitostí. Jediný typ transakce poskytuje jednoduchou, opakovatelnou jednotku práce a je navržen tak, aby vykonával klíčové operace systému OLTP.

Benchmark TPC-A měří, kolik transakcí za sekundu může systém provést, když je řízen z více terminálů. Zatímco TPC-A nespecifikuje počet terminálů, vyžaduje, aby společnosti škalovaly (zvyšovaly nebo snižovaly) počet terminálů podle propustnosti. TPC-A lze provozovat v konfiguraci rozlehlé nebo místní sítě s výkonem popsáním dvěma metrikami „místní propustnost TPC-A“ a „široká propustnost TPC-A“, měřené v transakcích za sekundu (tps). Tyto dvě metriky se vzájemně liší a nelze je tedy porovnávat. Specifikace benchmarku TPC-A vyžaduje, aby společnosti zveřejnily plné podrobnosti o svém srovnávacím testu, konfiguraci systému a nákladech na něj (včetně nákladů na údržbu po dobu 5 let).[58]

6.5 TPC-C

TPC-C je OLTP benchmark pro zpracování on-line transakcí schválený a vydaný v červenci 1992. Tento benchmark je složitější než předchozí benchmarky OLTP (jako např. TPC-A), má více typů transakcí, používá složitější databáze a celkovou strukturu provádění. TPC-C zahrnuje kombinaci pěti souběžných transakcí různých typů a složitostí, které jsou spuštěny on-line, nebo ve frontě pro odložené provedení. Databáze se skládá z devíti typů tabulek se širokou škálou záznamů a různých velikostí. TPC-C se měří v transakcích za minutu (tpm). Referenční hodnota představuje zatížení u velkoobchodního dodavatele. TPC-C není omezena na činnost žádného konkrétního obchodního segmentu, ale má zastupovat jakékoli odvětví, které se zabývá řízením, prodejem nebo distribucí produktů nebo služeb.[59]

6.6 Model TPC-C

Jako systémový benchmark OLTP simuluje TPC-C kompletní prostředí, kde skupina operátorů (terminálů) provádí transakce s databází. Benchmark se soustředí kolem hlavních činností (transakcí) v prostředí zadávání objednávek. Tyto transakce zahrnují zadání a dodání objednávek, evidenci plateb, kontrolu stavu objednávek a sledování stavu zásob na skladech.

V obchodním modelu TPC-C působí velkoobchodní dodavatel dílů (dále nazývaný Společnost), který má řadu skladů a k nim přidružených prodejních oblastí. Benchmark TPC-C je navržen tak, aby se škáloval stejně, jak společnost expanduje a vznikají nové sklady. Při škálování benchmarku však musí být zachovány určité konzistentní požadavky. Každý sklad v modelu TPC-C musí zásobovat 10 prodejních oblastí a každá oblast obsluhuje 3 000 zákazníků. Operátor z prodejní oblasti si může kdykoli vybrat jednu z pěti operací nebo transakcí nabízených systémem zadávání objednávek společnosti. Stejně jako samotné transakce je i frekvence jednotlivých transakcí modelována podle realistických scénářů.

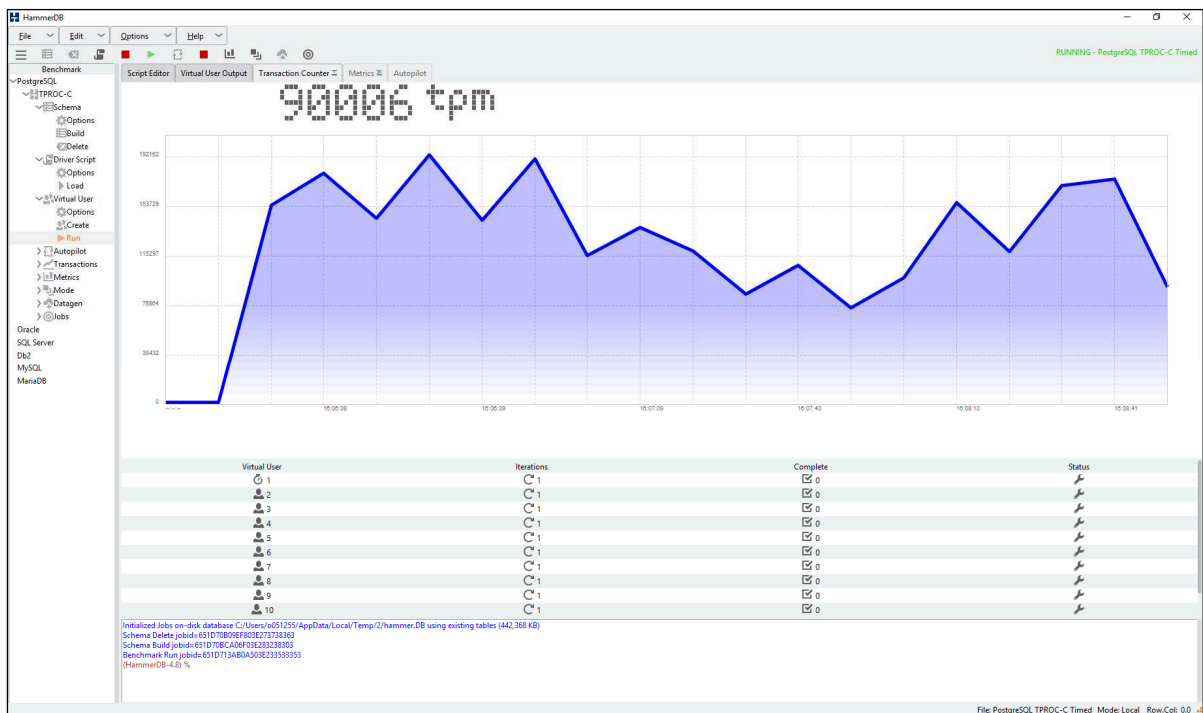
Nejčastější transakce spočívá v zadání nového příkazu, který se v průměru skládá z deseti různých položek. Každý sklad udržuje zásoby pro 100 000 položek v katalogu Společnosti a vyřizuje objednávky z těchto zásob. Ve skutečnosti však konkrétní sklad pravděpodobně nebude mít všechny díly potřebné k vyřízení každé objednávky. Proto TPC-C vyžaduje, aby téměř 10 % všech objednávek bylo dodáváno z jiného skladu Společnosti. Další častou transakcí je zaznamenání platby přijaté od zákazníka. Méně často budou operátoři požadovat stav dříve zadané objednávky, zpracovat dávku dodání deseti objednávek nebo se budou dotazovat systému na potenciální nedostatek stavu zásob v místním skladu. Celkem tedy pět typů transakcí, které se používají k modelování obchodní činnosti. Metrika výkonu hlášená TPC-C měří počet objednávek, které lze plně zpracovat za minutu, a je vyjádřena v tpm.

TPC-C byl navržen tak, aby obsahoval charakteristiky TPC-A, standardní verze DebitCredit TPC. Proto TPC-C zahrnuje všechny součásti základního benchmarku OLTP. Aby byl benchmark použitelný pro systémy s různými výpočetními výkony, implementace TPC-C škáluje jak počet terminálů, tak velikost databáze úměrně výpočetnímu výkonu měřeného systému. Aby bylo možné otestovat, zda je měřený systém plně připravený na provoz s dostatečnými schopnostmi obnovy, musí databáze poskytovat vlastnosti definované jako ACID: atomicitu, konzistenci, izolaci a trvanlivost. Pro nezávislé ověření výsledků srovnávacích testů, musí zadavatel testu zveřejnit všechny informace nezbytné k nahlášené výkonnosti. Tento výkon, který měří propustnost systému, musí být zveřejněn spolu s celkovými náklady na systém. Celkové náklady na systém jsou blízkou aproximací skutečných nákladů na část systému dodanou dodavatelem pro koncového uživatele. Zahrnuje náklady na všechny hardwarové a softwarové komponenty, náklady na údržbu po dobu 5 let a dostatečnou úložnou kapacitu pro uložení dat generovaných po dobu 180 osmihodinových dnů provozu při uváděné propustnosti.[60]

6.7 HammerDB

HammerDB je přední benchmarking a zátěžový testovací software pro celosvětově nejoblíbenější databáze podporující Oracle Database, Microsoft SQL Server, IBM Db2, MySQL, MariaDB a PostgreSQL. HammerDB vytvoří testovací schéma, naplní jej daty a simuluje zatížení více virtuálními uživateli pro transakční i analytické scénáře. Výsledky se pak dají použít k odvození smysluplných informací o použitém prostředí, jako je např. porovnání výkonu hardwaru a konfigurace softwaru.

Software HammerDB (jehož grafické prostředí je vidět na Obr. 4) je využíván všemi předními databázovými vývojáři a technologickými společnostmi. Byl stažen více než sto tisíckrát ve více než 180 zemích světa. Mezi společnostmi, využívající tento software, patří např. Oracle, IBM, Intel, Dell/EMC HPE, Huawei, Lenovo a mnoho dalších.[61]



Obr. 4 – Prostředí software HammerDB

6.8 TPROC-C

TPROC-C je úloha OLTP implementovaná v HammerDB odvozená ze specifikace TPC-C s úpravami, které usnadňují a zefektivňují provoz HammerDB v libovolném podporovaném databázovém prostředí. Úloha HammerDB TPROC-C je open-source úloha odvozená ze standardu TPC-C Benchmark Standard a jako taková není srovnatelná s publikovanými výsledky TPC-C, protože výsledky jsou v souladu s podmnožinou, nikoli s úplným standardem srovnávacího testu TPC-C. Název úlohy HammerDB TPROC-C znamená Transaction Processing Benchmark derived from the TPC-C specification (srovnávací test zpracování transakcí odvozený ze specifikace TPC-C).[61]

7 Praktická část

Bylo provedeno porovnání relativního výkonu tří databázových systémů PostgreSQL, SQL Server a MySQL pomocí open-source nástroje HammerDB. První dva z těchto databázových systémů jsou mj. používány v organizaci státní správy (kde pracuji) pro provoz tzv. kritických aplikací, a i když je vnitropodnikovou politikou stále preferován systém společnosti Microsoft, tak už několik aplikací „běhá“ na open-source databázi, což je dáno její podporou a využíváním dodavatelů těchto aplikací. To dokládá výše popsanou oblíbenost PostgreSQL.

7.1 Testovací hardware a software

Pro srovnávací test byl použit virtuální server (technologie Hyper-V) od společnosti Microsoft s parametry uvedenými v Tab. 12.

Tab. 12 – Parametry testovaného serveru

Operační systém	Windows Server 2022 Standard, verze 21H2
Procesor	Intel(R) Xeon(R) Gold 6248 CPU @2.49 GHz (4 procesory)
Paměť (RAM)	16,00 GB
Typ systému	64-bitový operační systém na x64 procesoru

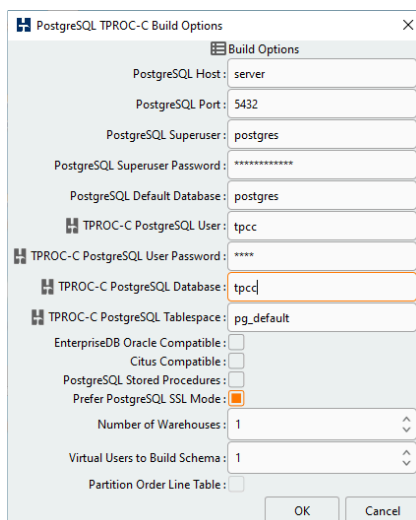
Na tento virtuální server byly dále nainstalovány databázové systémy Microsoft SQL Server 2022 Standard Edition, PostgreSQL 16.1 a MySQL 8.2. Aby test jednotlivých systémů nebyl ovlivněn během dalších dvou systémů, byly vždy tyto systémy vypnuté (zastaveny příslušné služby). Standardně SQL Server využívá port 1433, PostgreSQL port 5432 a MySQL port 3306. Zvláště SQL Server si alokuje tolik operační paměti RAM, kolik je nastaveno v Properties serveru v položce Server memory option, i když není tato kapacita v dané chvíli plně využita. Instalace byly provedeny pomocí tzv. MSI balíčků a byla ponechána defaultní nastavení a byla zvolena silná hesla k administrátorským účtům. Dále byl nainstalován software HammerDB 4.9 a nakonfigurován (podle návodu na oficiálních webových stránkách software) pro použití databázové systémy.

Nástroj HammerDB může spouštět srovnávací testy s instalacemi Oracle, PostgreSQL, SQL Server, MySQL, MariaDB a DB2. Generuje srovnávací hodnoty ve formě TPM (transactions placed per minute – transakce provedené za minutu) a OPM (orders placed per minute – objednávky provedené za minutu) na testovací cyklus a testuje výkon konfigurace instance a infrastruktury na níž „běží“. Syntetickou datovou sadu úlohy vytváří pomocí vnitřního mechanismu nástroje nebo je možné ji automatizovat pomocí tzv. Autopilota. Umí tedy porovnat relativní výkon mezi dvěma instancemi databázových serverů. Uživatel načte stejná testovací data do dvou různých instancí, spustí stejný test na každém serveru a porovná výsledky. Porovná se hrubý výkon serveru a lze identifikovat rozdíly ve výkonu serveru.[62]

7.2 Napojení PostgreSQL

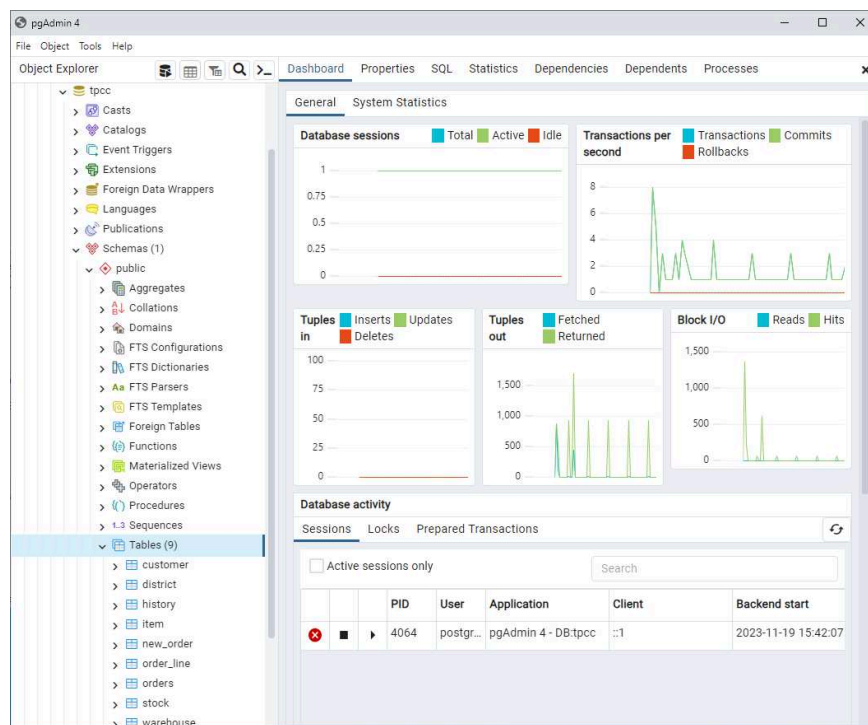
Pokud je HammerDB nainstalován na klientské stanici a PostgreSQL na testovacím serveru je nutné zkopírovat některé další knihovny databázového systému do podadresáře `..\lib\pgtcl2.1.1`, instalačního adresáře benchmarku. V tomto adresáři je pouze základní knihovna `libpgtcl.dll`, která vyžaduje další. Potřebné knihovny, lze zjistit např. pomocí nástroje `Dependencies`, který ukáže závislost základní knihovny na dalších vyžadovaných.

HammerDB k databázovému systému poté přistupuje pomocí nastavení Options viz Obr. 5, kde jsou uvedeny přístupové údaje jako název serveru, port, uživatel, heslo a další. Superuser a jeho heslo jsou vytvořeny při instalaci databázového systému.



Obr. 5 – Nastavení připojení k PostgreSQL

Funkční konfiguraci je možné otestovat pomocí funkce Build, která v PostgreSQL úspěšně vytvoří testovací databázi TPCC (název lze samozřejmě změnit). Tato funkce vytvoří testovací databázi a v ní příslušné tabulky, indexy a funkce. Vytvořenou databázi, včetně jejích tabulek, je možné si prohlédnout např. pomocí aplikace pgAdmin (ta je součástí instalace PostgreSQL) viz Obr. 6 nebo DBeaver, které umožňují správu databáze pomocí grafického rozhraní API.

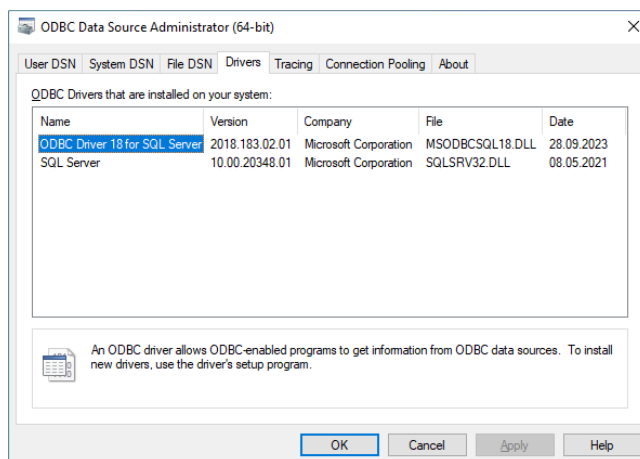


Obr. 6 – Grafické rozhraní pgAdmin s vytvořenou testovací databází

7.3 Napojení SQL Server

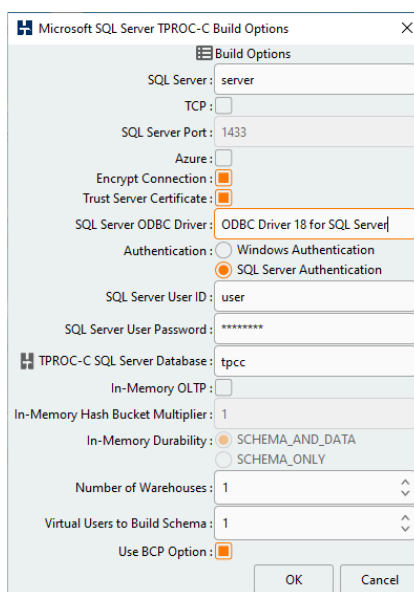
Pro napojení HammerDB na SQL Server je možné použít ověřování pomocí systému Windows nebo ověřování pomocí SQL Serveru (uživatel vytvořený v sekci Security instance SQL serveru).

HammerDB umožňuje použití obou metod, ale je nutné mít nastavenou odpovídající konfiguraci na SQL Server nebo Mixed mode, který umožňuje použití obou metod ověřování. Kromě toho zvolená metoda ověřování musí být kompatibilní s vybraným ovladačem ODBC, který je nutné nainstalovat (instalační MSI balíček lze stáhnout ze stránek společnosti Microsoft) a bez kterého není možné se k SQL Server připojit. Dostupné ovladače lze v operačním systému Windows zjistit pomocí nástroje Správce zdrojů dat ODBC viz Obr. 7.



Obr. 7 – Správce zdrojů dat ODBC

Název ovladače musí být zadán do HammerDB přesně tak, jak je uvedeno ve Správci zdrojů dat. Výchozí hodnota je „ODBC Driver 18 for SQL Server“ viz Obr. 8. Konfiguraci lze otestovat stejně jako u PostgreSQL.

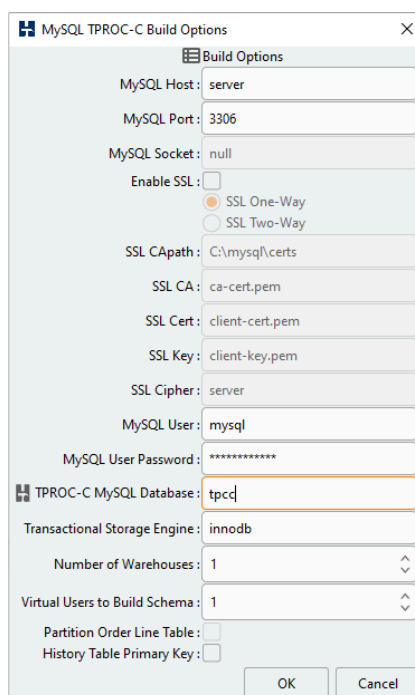


Obr. 8 – Nastavení připojení k SQL Serveru

7.4 Napojení MySQL

Podobně jako u PostgreSQL, pokud je HammerDB nainstalován na klientské stanici a MySQL na testovacím serveru, je potřeba zkopírovat další knihovny databázového systému do podadresáře `..\lib\mysqltcl3.052libmysqltcl.dll`, instalačního adresáře benchmarku, kde je jen základní knihovna `libmysqltcl.dll`.

HammerDB k databázovému systému poté přistupuje, stejně jako u PostgreSQL, pomocí nastavení Options viz Obr. 9, kde jsou uvedeny přístupové údaje jako název serveru, port, uživatel, heslo a další. User a jeho heslo jsou vytvořeny při instalaci databázového systému.



Obr. 9 – Nastavení připojení k MySQL

7.5 Uložení konfigurace

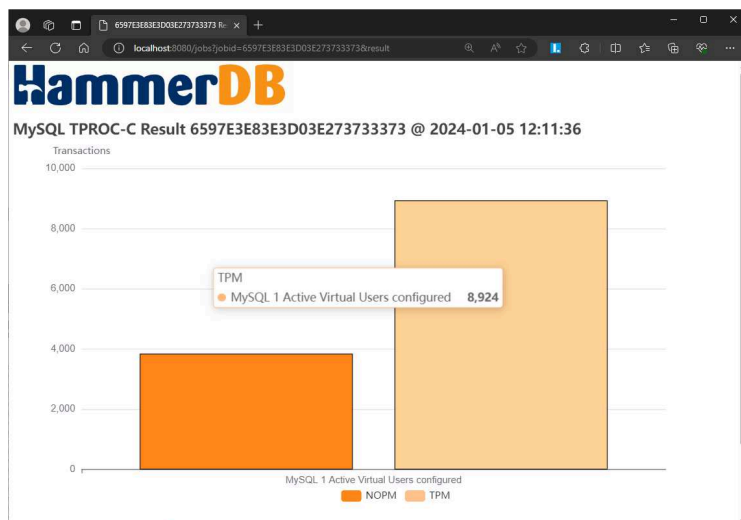
Aby nebylo nutné po každém spuštění HammerDB konfigurovat jednotlivá připojení na testované databázové systémy a nastavení Autopilota, (který byl pro testování použit) je možné „ručně“ upravit konfigurační soubory .xml v podadresáři ..\config\. Tyto soubory jsou pojmenovány intuitivně, každý podle příslušného databázového systému. Pro nastavení konfigurace Autopilota je to soubor generic.xml v témže podadresáři.

7.6 Spuštění a automatizace testů

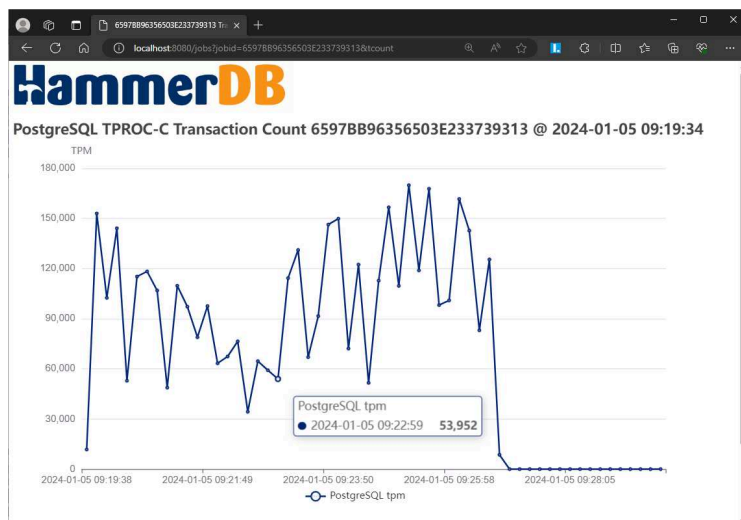
Pro spuštění testů nad jednotlivými databázovými systémy je nejprve nutné vytvořit testovací databázi (její název lze změnit, ale byl ponechán defaultní název TPCC) a po ukončení testu ji zase smazat. Vše „zařídí“ funkce Build a Delete v menu Options. Po úspěšném vytvoření testovací databáze (schématu) lze v menu Virtual Users ručně vytvořit požadovaný počet uživatelů (pomocí funkce Create) a spustit test (pomocí funkce Run). Testy lze zautomatizovat pomocí Autopilota, kde lze nastavit počet připojených uživatelů a čas testu pro tento počet uživatelů. Je tedy možné testovat, jak se systém bude chovat např. pro 1, 5, 10, 25, 50 a 100 připojených uživatelů, jako v případě testů v praktické části bakalářské práce. Autopilot po startu (funkce Autopilot v menu Autopilot) spustí test pro jednoho uživatele po stanovenou dobu a zapíše výsledek do interní databáze benchmarku (nikoliv do testované). Poté spustí test znovu s pěti uživateli po stanovenou dobu a opět zapíše výsledek. A takto se to opakuje pro všechny naplánované uživatele. Není tedy potřeba jednotlivé testy spouštět „ručně“, každý zvlášť pro určitý počet předem vytvořených připojených uživatelů.

HammerDB, jak už bylo popsáno výše, si výsledky spuštěných testů a všech provedených kroků zapisuje do interní databáze. Dále má integrovanou webovou službu, kterou lze z menu Option

spustit, a pomocí webového prohlížeče lze výsledky testů (naměřené hodnoty OPM a TPM) a dalších kroků zobrazit. Po kliknutí na danou úlohu (job) lze zobrazit details, jako log průběhu testu, typ testu, databázový systém, konfigurace testu, výsledek (viz Obr. 10), status testu, graf s počtem transakcí v průběhu testu (viz Obr. 11), časové razítko a možnost smazání daného testu. HammerDB má pro každý typ podporovaného databázového systému vyhrazenou jednu barvu. Pro PostgreSQL je to modrá, SQL server žlutá a MySQL oranžová. Toto barevné rozlišení bylo dodrženo i pro tabulky a grafy s výsledky testů praktické části.



Obr. 10 – Výsledky počtu objednávek a počtu transakcí MySQL při 9. měření



Obr. 11 – Graf transakcí testu PostgreSQL při 9. měření

7.7 Vlastní měření

Jak již bylo popsáno výše, pro vlastní měření benchmarku byly nainstalovány tři popisované databázové systémy na samostatný virtuální server. Na klientskou stanici ve stejné síti byl nainstalován benchmarkový software HammerDB a provedena jeho konfigurace, zejména instalace potřebných knihoven a nastavení přístupových údajů (server, port, uživatel a heslo). Dále byl nad každým databázovým systémem proveden test konfigurace, vytvořením testovací databáze, spuštěním testu a smazáním testovací databáze. Takto otestované, byly přístupové údaje uloženy do příslušných konfiguračních souborů, aby je nebylo nutné před každým testem znovu nastavovat. Dále byl nastaven Autopilot, kdy defaultní nastavení bylo změněno tak, aby zátěžový test postupně

proběhl pro 1, 5, 10, 25, 50 a 100 uživatelů. Toto nastavení bylo opět uloženo do příslušného konfiguračního souboru. Doba běhu testu byla ponechána na 5 minut a spuštění testů po 10 minutách. Nastavení Autopilotu bylo nad každým databázovým systémem otestováno. U PostgreSQL bylo nutné upravit parametr `max_connections = 150`, v konfiguračním souboru `postgresql.conf` (hlavní konfigurační soubor tohoto systému). Původní hodnota parametru byla 100, ale při spuštění testu se sto uživateli, test zhavaroval, protože HammerDB potřebuje ještě další připojení pro komunikaci. Po tomto odladění již bylo vše připravené pro vlastní měření.

Protože měření pro každý databázový systém trvalo více než hodinu (start příslušné služby, vytvoření testovací databáze, vlastní měření 6×10 minut, smazání databáze a stop služby), byla měření prováděna jednou denně v pořadí PostgreSQL, SQL Server a MySQL v průběhu 10 dní. Výsledkem je tedy 10 měření pro každý databázový systém při všech testovaných uživatelích. Na Obr. 12 je souhrn jednoho (posledního) měření, zobrazený ve webovém prohlížeči, jak bylo popsáno výše. Jsou zde vidět jednotlivé kroky, jejich status, časy a i (hvězdičkou) označen nejlepší výsledek.

Jobid	Database	Benchmark	Date	Workload	Status
659BAA4F59E403E263034333	PostgreSQL	TPROC-C	2024-01-08 08:54:55	Schema Build	✓
659BAAF5BD2603E293731323	PostgreSQL	TPROC-C	2024-01-08 08:57:41	Benchmark Run	✓
659BAD4F29CD03E263332303	PostgreSQL	TPROC-C	2024-01-08 09:07:43	Benchmark Run	✓
659BAFAB8C0803E233939393	PostgreSQL	TPROC-C	2024-01-08 09:17:47	Benchmark Run	✓
659BB20AF59103E253837253	PostgreSQL	TPROC-C	2024-01-08 09:27:54	Benchmark Run	✓
659BB470639D03E223433383	PostgreSQL	TPROC-C	2024-01-08 09:38:08	Benchmark Run	✓
659BB6E3D95703E293131303	PostgreSQL	TPROC-C	2024-01-08 09:48:35	Benchmark Run	✓
659BBF41D61703E243831393	PostgreSQL	TPROC-C	2024-01-08 10:24:17	Schema Delete	✓
659BBF71F27403E23031303	MSSQLServer	TPROC-C	2024-01-08 10:25:05	Schema Build	✓
659BC081951803E293430313	MSSQLServer	TPROC-C	2024-01-08 10:29:37	Benchmark Run	✓
659BC20BF8B303E243131373	MSSQLServer	TPROC-C	2024-01-08 10:39:39	Benchmark Run	✓
659BC53763AC03E243239393	MSSQLServer	TPROC-C	2024-01-08 10:49:43	Benchmark Run	✓
659BC796CD4A03E243236303	MSSQLServer	TPROC-C	2024-01-08 10:59:50	Benchmark Run	✓
659BC9FC3B9B03E213339353	MSSQLServer	TPROC-C	2024-01-08 11:10:04	Benchmark Run	★
659BC70B18803E223733383	MSSQLServer	TPROC-C	2024-01-08 11:20:32	Benchmark Run	✓
659BD9E2B5A103E243135393	MSSQLServer	TPROC-C	2024-01-08 12:17:54	Schema Delete	✓
659BD711AE603E293383333	MySQL	TPROC-C	2024-01-08 12:24:33	Schema Build	✓
659BD94504F03E213231383	MySQL	TPROC-C	2024-01-08 12:29:24	Benchmark Run	✓
659DED86A603E213731343	MySQL	TPROC-C	2024-01-08 12:39:25	Benchmark Run	✓
659BE1491EA703E243132393	MySQL	TPROC-C	2024-01-08 12:49:29	Benchmark Run	✓
659BE3A7881803E273034383	MySQL	TPROC-C	2024-01-08 12:59:35	Benchmark Run	✓
659BE60FF6F903E283931393	MySQL	TPROC-C	2024-01-08 13:09:51	Benchmark Run	✓
659BE826CB03E263632313	MySQL	TPROC-C	2024-01-08 13:20:18	Benchmark Run	✓
659BE8A44AC303E263739363	MySQL	TPROC-C	2024-01-08 13:33:40	Schema Delete	✓

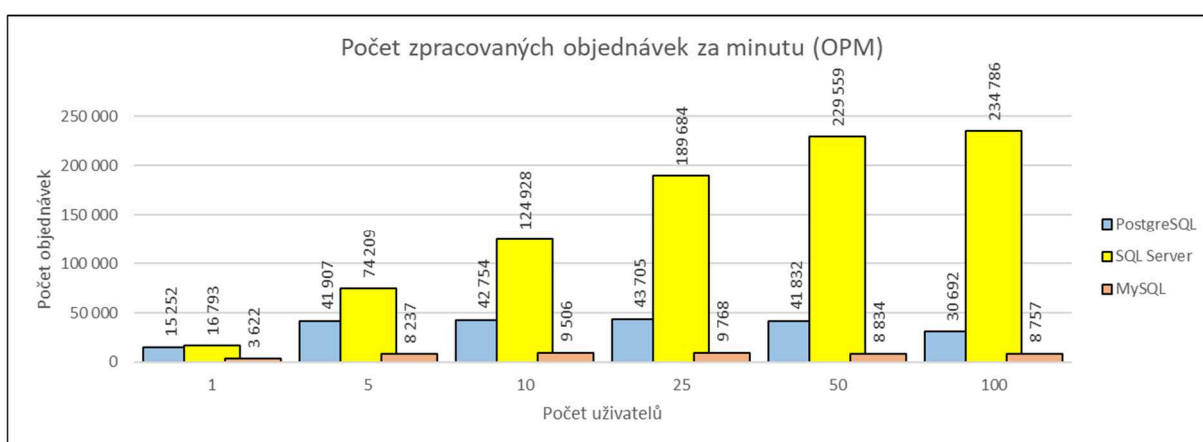
Obr. 12 – Souhrn 10. měření zobrazený ve webovém prohlížeči

Výsledky jednotlivých testů nelze za všechna provedená měření zobrazit souhrnně (je možné „rozkliknout“ daný test a zobrazit výsledek), tak byly výsledky jednotlivých měření přepisovány do předem připravených Excelovských tabulek. V tabulkách jsou uvedeny datum měření, čas spuštění každého testu a výsledky: počet zpracovaných objednávek za minutu (OPM) a počet zpracovaných transakcí za minutu (TPM). Pro lepší přehlednost (a rychlejší možnost porovnání) naměřených výsledků byly následně vytvořeny dva sloupcové grafy každého měření. Jeden pro zpracované objednávky a druhý pro zpracované transakce. Protože měření probíhala v reálném provozu a zatížení sítě, jsou i výsledky ovlivněny (pro objektivní zhodnocení výkonnosti, ku prospěchu) např. momentálním vytížením IT infrastruktury. První měření byla prováděna na konci roku 2023, kdy probíhaly např. uzávěrkové práce, inventury apod., a další měření v prvním týdnu roku 2024. Výsledky (aritmetický průměr všech naměřených hodnot) jsou uvedené v Tab. 13. Tabulky a grafy všech jednotlivých 10. měření jsou poté přílohami bakalářské práce, zde jsou tedy uvedené aritmetické průměry výsledků těchto měření.

Tab. 13 - Aritmetický průměr (počet zpracovaných) OPM a TPM jednotlivých systémů všech 10. provedených měření

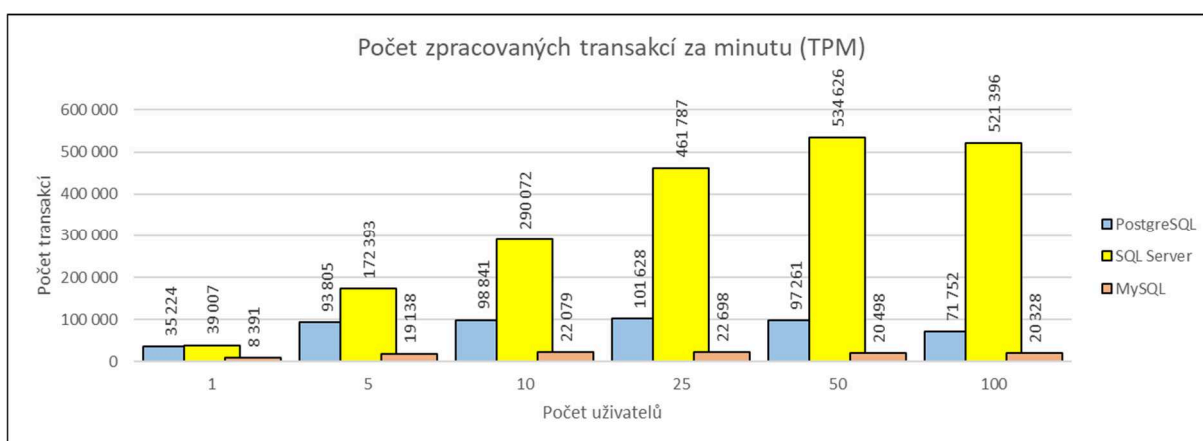
Aritmetický průměr všech měření	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	OPM	15 252	41 907	42 754	43 705	41 832	30 692
	TPM	35 224	93 805	98 841	101 628	97 261	71 752	
SQL Server	OPM	16 793	74 209	124 928	189 684	229 559	234 786	
	TPM	39 007	172 393	290 072	461 787	534 626	521 396	
MySQL	OPM	3 622	8 237	9 506	9 768	8 834	8 757	
	TPM	8 391	19 138	22 079	22 698	20 498	20 328	

Grafické zobrazení všech naměřených hodnot OPM z Tab. 13 je uvedeno v Graf 1. Na vodorovné ose x grafu je počet uživatelů (1, 5, 10, 25, 50 a 100) a na svislé ose y počet zpracovaných objednávek. Jak již bylo popsáno výše, pro každý testovaný systém byla v tabulce a grafech zvolena barva, kterou danému databázovému systému „přidělil“ HammerDB. Pro PostgreSQL je to modrá, SQL server žlutá a MySQL oranžová.



Graf 1 – Počet OPM pro jednotlivé systémy (aritmetický průměr všech 10. provedených měření)

Grafické zobrazení všech naměřených hodnot TPM z Tab. 13 je uvedeno v Graf 2. Na vodorovné ose x grafu je opět počet uživatelů (1, 5, 10, 25, 50 a 100) a na svislé ose y počet provedených transakcí.



Graf 2 – Počet TPM pro jednotlivé systémy (aritmetický průměr všech 10. provedených měření)

7.8 Vyhodnocení

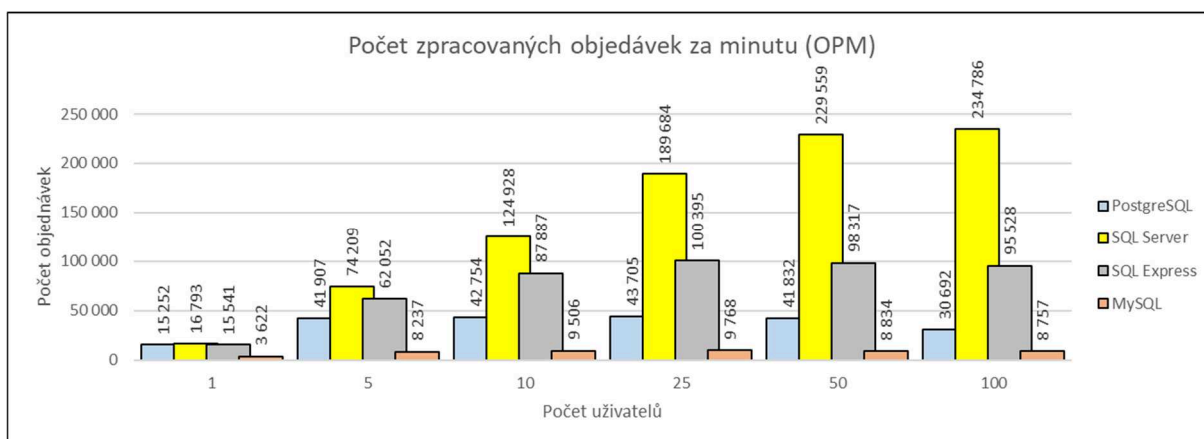
Z obou grafů je již na první pohled vidět výkonnostní rozdíl mezi jednotlivými systémy. SQL Server dokáže zpracovat více objednávek i transakcí, a to zvláště při vyšším počtu připojených uživatelů. Protože SQL Server je (narozdíl od dalších dvou porovnávaných systémů) placený

software, bylo měření (mj. i z důvodu akademické zvědavosti autora) nakonec provedeno i na SQL Express, což je neplacená verze SQL Serveru s omezeními (maximálně 1 GB využívané operační paměti, maximálně 10 GB velikost databáze, maximálně 1 MB vyrovnávací paměti, menší z možností jedno CPU nebo čtyři jádra, počet připojení je bez omezení [63]). I zde proběhlo 10 měření během čtyř dnů (první tři dny 3 měření a čtvrtý den 1) a výsledky měření (aritmetické průměry) jsou uvedeny v Tab. 14, která je rozšířením Tab. 13 o SQL Express a grafické zpracování výsledků poté uvedeno v Graf 3 a v Graf 4.

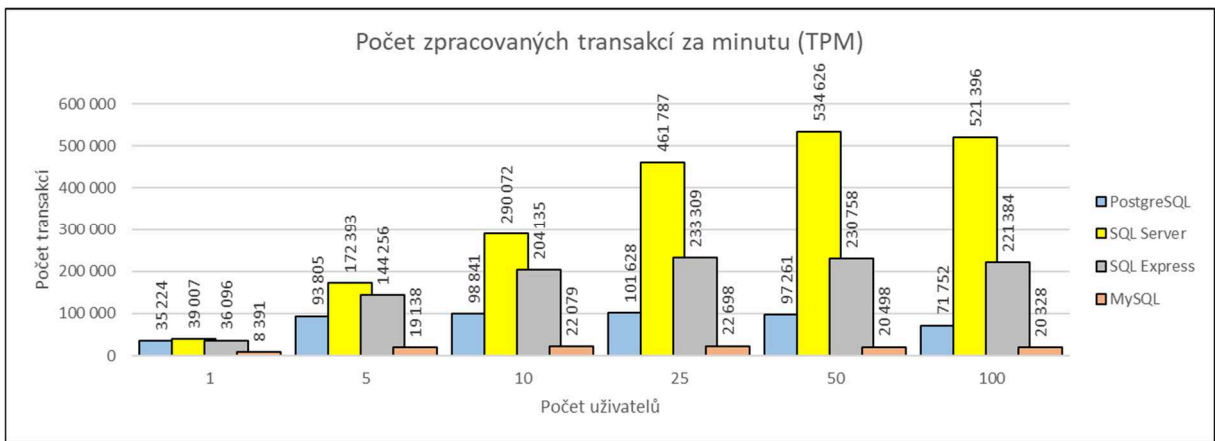
Tab. 14 - Aritmetický průměr OPM a TPM, rozšíření Tab. 13 o SQL Express

Aritmetický průměr všech měření		Počet uživatelů		1	5	10	25	50	100
		PostgreSQL	OPM	15 252	41 907	42 754	43 705	41 832	30 692
		TPM	35 224	93 805	98 841	101 628	97 261	71 752	
		SQL Server	OPM	16 793	74 209	124 928	189 684	229 559	234 786
			TPM	39 007	172 393	290 072	461 787	534 626	521 396
		SQL Express	OPM	15 541	62 052	87 887	100 395	98 317	95 528
			TPM	36 096	144 256	204 135	233 309	230 758	221 384
		MySQL	OPM	3 622	8 237	9 506	9 768	8 834	8 757
			TPM	8 391	19 138	22 079	22 698	20 498	20 328

I tato „light“ verze SQL Serveru ostatní dva systémy, výkonnostně převyšuje. Pokud pomineme licenční podmínky, popsané v rešeršní části, které použití omezují zejména při komerčním využití, je možné pro malý e-shop s návštěvností od několika až po stovky zákazníků denně, použít systém MySQL, který výkonnostně požadavky pokryje (při současném připojení 100 uživatelů je schopný vyřídit přes 8 000 objednávek během minuty). Pro velký e-shop s denní návštěvností tisíců zákazníků, by bylo vhodné využít PostgreSQL. Pro e-shop velkoobchodní společnosti, nabízející velké množství položek, který musí pokrýt požadavky desítek tisíc zákazníků, by bylo dobré už využít produkt společnosti Microsoft. Dále je možné vidět, jak se zvyšujícím se počtem připojených uživatelů se zvyšuje počet zpracovaných objednávek a transakcí. Zatímco u PostgreSQL a MySQL, se už u 5 a více uživatelů tento trend zastaví u obou verzí SQL Serveru je to až u 50 uživatelů.



Graf 3 – Počet zpracovaných OPM, rozšíření o SQL Express



Graf 4 – Počet zpracovaných TPM, rozšíření o SQL Express

8 Závěr

V teoretické části byly popsány a porovnány tři vybrané databázové systémy a jejich specifické vlastnosti ze zdrojů uvedených v části Seznam použitých zdrojů, tj. odborných textů výrobců nebo distributorů těchto systémů a textů zabývajících se IT technologiemi. Dále zde byly popsány zátěžové (benchmarkové) testy, pomocí kterých lze zjistit relativní výkon těchto systémů nasazených v určitém prostředí (ať už se jedná o virtuální server nebo cloudové prostředí) a nástroj HammerDB, který tyto testy sofistikovaně a automatizovaně provádí. Na základě těchto získaných informací pro běžnou aplikaci (např. interní aplikace pro zaměstnance Finanční správy, kterých je kolem 13 000, případně e-shop, jakých se v covidové době objevilo mnoho) by jako nejvhodnější mohl být použit systém PostgreSQL, a to zejména z důvodu jeho nulových pořizovacích nákladů (myšleno samotného software). Dále jsou to jeho licenční podmínky, kdy není nutné řešit žádná omezení. U SQL Server je nutné řešit např. jádra procesoru nebo počet přistupujících uživatelů (dle toho se odvíjí i pořizovací cena). Microsoft sice nabízí i neplacenou verzi SQL Express, ale zde je limitující zejména velikost použité databáze a tím i množství v ní uložených dat. U MySQL je pro komerční využití nutno zakoupit některou z placených licencí. Otázkou zůstává, zda interní aplikace používaná zaměstnanci a vyvinutá vlastními silami je komerční produkt, takže je nutné prostudovat licenční podmínky. Výhodou PostgreSQL je i možnost instalace na různých platformách, a tak není potřeba pořizovat i placený operační systém. U placených systémů je to zase poskytovaná technická podpora ze strany dodavatele a případný problém není nutné řešit svépomocí hledáním na diskusních fórech apod.

V praktické části byl měřen relativní výkon vybraných databázových systémů (PostgreSQL, SQL Server a MySQL) pomocí benchmarkového nástroje HammerDB. Všechny tři systémy podporují jazyk SQL, což dokládá i použití zkratky v jejich názvech. Původním záměrem praktické části bylo pomocí SQL skriptu tohoto jazyka vytvořit v každém systému tabulku o velkém množství záznamů (např. 10 milionů) a položek se základními datovými typy (text, celé a desetinné číslo, datum apod.). Dále tuto tabulku naplnit daty, data následně zeditovat a nakonec vymazat. Pomocí změřených časů provedení jednotlivých příkazů porovnat výkon systémů. Software HammerDB, kde testy výkonnosti probíhají mnohem sofistikovaněji a měření je možné zautomatizovat, práci ulehčil a vzhledem k tomu, že software je velmi uznávaným nástrojem pro měření výkonu databázových systémů, i poskytnuté výsledky jsou mnohem více vypovídající. Instalace software a prvotní nastavení (zvláště, kdy bylo nutné nastudovat uživatelskou dokumentaci o napojení různých databází) bylo i příjemně stráveným časem a získáním nových poznatků, které bude možné využít i v mém zaměstnání. Pro test výkonu byl vybrán model TPC-C, který simuluje velkoobchodního dodavatele dílů a zahrnuje zadání a dodání objednávek, evidenci plateb, kontrolu stavu objednávek a sledování stavu zásob na skladech. Princip modelu je popsán v rešeršní části práce. Výsledkem je počet zpracovaných objednávek za minutu (OPM) a počet zpracovaných transakcí za minutu (TPM). Z naměřených hodnot vyplývá, že největší výkon poskytuje databázový systém společnosti Microsoft SQL Server, který během minuty při 100 uživateli, provedl přes 500 000 transakcí. Jedná se sice o placený produkt, jehož pořízení není levnou záležitostí, ale oproti svým konkurentům, dokázal zpracovat násobně vyšší počet požadavků v daném čase. Otázkou ovšem zůstává, zda pro běžné aplikace provozu je vždy až takový výkon potřeba. I druhý systém v pořadí (PostgreSQL), který je zdarma, poskytl na stejném hardware výkon přes 70 000 transakcí za minutu při 100 uživateli. Při menším počtu uživatelů je to i 100 000 transakcí. Třetí v pořadí je poté MySQL (kolem 20 000 transakcí za

minutu při 100 uživatelích). Testovaná verze je zdarma, ale je možné ji použít jen pro nekomerční účely (tzn. že při provozování např. e-shopu by byly porušeny licenční podmínky). Pro komerční použití je nutné zakoupit jednu z placených verzí, která umožní i mnohem vyšší výkon.

Z důvodu objektivnosti bylo nakonec provedeno měření i na bezplatném SQL Express. Těchto měření bylo nakonec provedeno také 10, ale 230 000 transakcí (při 50 uživatelích) ji i tak zařadilo na 1. místo výkonnosti.

Seznam použitých zdrojů

- [1] *Databáze – Wikisofia* [online]. [cit. 2023-11-17]. Dostupné z: <https://wikisofia.cz/wiki/Datab%C3%A1ze>
- [2] *Základní informace o databázích - Podpora Microsoftu* [online]. [cit. 2022-11-15]. Dostupné z: <https://support.microsoft.com/cs-cz/office/z%C3%A1kladn%C3%AD-informace-o-datab%C3%A1z%C3%ADch-a849ac16-07c7-4a31-9948-3c8c94a7c204>
- [3] *Relační databáze vs. nerelační databáze: Čím se liší? | MasterDC* [online]. [cit. 2022-11-15]. Dostupné z: <https://www.master.cz/blog/relacni-databaze-nerelacni-databaze-jake-jsou-rozdily/>
- [4] *Relační databáze Pojem databáze, druhy databází* [online]. nedatováno [cit. 2022-11-15]. Dostupné z: http://en.wikipedia.org/wiki/Oracle_database
- [5] *Hierarchický model v DBMS - GeeksforGeeks* [online]. [cit. 2023-11-19]. Dostupné z: https://www-geeksforgeeks-org.translate.goog/hierarchical-model-in-dbms/?_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs
- [6] *Network Model in DBMS - GeeksforGeeks* [online]. [cit. 2023-11-19]. Dostupné z: <https://www.geeksforgeeks.org/network-model-in-dbms/>
- [7] *Difference between RDBMS and OODBMS - GeeksforGeeks* [online]. [cit. 2023-11-19]. Dostupné z: <https://www.geeksforgeeks.org/difference-between-rdbms-and-oodbms/>
- [8] *Co je strukturovaný dotazovací jazyk (SQL)?* [online]. [cit. 2022-11-15]. Dostupné z: https://www-techartget-com.translate.goog/searchdatamanagement/definition/SQL?_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs
- [9] *CzWiki > Tabulka (databáze)* [online]. [cit. 2022-11-22]. Dostupné z: [https://czwiki.cz/Lexikon/Tabulka_\(datab%C3%A1ze\)](https://czwiki.cz/Lexikon/Tabulka_(datab%C3%A1ze))
- [10] *Využití databázových indexů - Root.cz* [online]. [cit. 2023-10-05]. Dostupné z: <https://www.root.cz/clanky/vyuziti-databazovych-indexu/>
- [11] *CzWiki > Pohled (databáze)* [online]. [cit. 2022-11-22]. Dostupné z: [https://czwiki.cz/Lexikon/Pohled_\(datab%C3%A1ze\)](https://czwiki.cz/Lexikon/Pohled_(datab%C3%A1ze))
- [12] *SQL Transakce COMMIT ROLLBACK - předcházejte průšvihům | BI Portál* [online]. [cit. 2022-11-22]. Dostupné z: <https://biportal.cz/sql-transaction-commit-rollback/>
- [13] *What is MVCC? How multiversion concurrency control works* [online]. [cit. 2023-11-17]. Dostupné z: <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/What-is-MVCC-How-does-Multiversion-Concurrency-Control-work>
- [14] *What is Online Transaction Processing (OLTP) | Oracle* [online]. [cit. 2023-09-19]. Dostupné z: <https://www.oracle.com/database/what-is-oltp/>
- [15] *Přehled technologie OLAP (Online Analytical Processing) - Podpora Microsoftu* [online]. [cit. 2023-10-02]. Dostupné z: <https://support.microsoft.com/cs-cz/office/p%C5%99ehled-technologie-olap-online-analytical-processing-15d2cdde-f70b-4277-b009-ed732b75fdd6>

- [16] *Třívrstvá architektura (Three-tier architecture) - ManagementMania.com* [online]. [cit. 2023-11-21]. Dostupné z: <https://managementmania.com/cs/trivrstva-architektura-three-tier-architecture>
- [17] *Třívrstvá architektura (Three-tier architecture) - ManagementMania.com* [online]. [cit. 2022-11-22]. Dostupné z: <https://managementmania.com/cs/trivrstva-architektura-three-tier-architecture>
- [18] *Licence počítačových programů a jejich typy | NaPočítači.cz* [online]. [cit. 2023-11-17]. Dostupné z: https://www.napocitaci.cz/33/licence-pocitacovych-programu-a-jejich-typy-uniqueidgOkE4NvrWuNY54vrLeM67_Sxz_UxTxoh9mbKTrAPMMA/
- [19] *Opensource.com* [online]. [cit. 2023-09-19]. Dostupné z: <https://opensource.com/>
- [20] *Secure Sockets Layers (SSL) - IBM Documentation* [online]. [cit. 2023-10-27]. Dostupné z: <https://www.ibm.com/docs/en/b2b-integrator/5.2?topic=522-secure-sockets-layers-ssl>
- [21] *What is a Data Warehouse? | Google Cloud* [online]. [cit. 2023-11-21]. Dostupné z: <https://cloud.google.com/learn/what-is-a-data-warehouse>
- [22] *What is Database Benchmark? Definition & FAQs | ScyllaDB* [online]. [cit. 2023-09-27]. Dostupné z: <https://www.scylladb.com/glossary/database-benchmark/>
- [23] MOMJIAN, Bruce. *PostgreSQL - praktický průvodce*. Brno: Computer Press a.s., 2003. ISBN 80-7226-954-2.
- [24] *What is PostgreSQL* [online]. [cit. 2023-09-18]. Dostupné z: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>
- [25] *PostgreSQL: Zásady verzování* [online]. [cit. 2023-09-19]. Dostupné z: https://www-postgresql-org.translate.goog/support/versioning/?_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs
- [26] *EDB Docs - Requirements* [online]. [cit. 2023-09-19]. Dostupné z: <https://www.enterprisedb.com/docs/supported-open-source/postgresql/installing/requirements/>
- [27] *Microsoft SQL Server – Wikipedie* [online]. [cit. 2022-11-22]. Dostupné z: https://en-m-wikipedia-org.translate.goog/wiki/Microsoft_SQL_Server?_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs
- [28] *Installation guidance for SQL Server on Linux - SQL Server | Microsoft Learn* [online]. [cit. 2023-09-20]. Dostupné z: <https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-setup?view=sql-server-ver16>
- [29] *Latest updates and version history for SQL Server - SQL Server | Microsoft Learn* [online]. [cit. 2023-09-20]. Dostupné z: <https://learn.microsoft.com/en-us/troubleshoot/sql/releases/download-and-install-latest-updates>
- [30] *SQL Server 2019: Hardwarové a softwarové požadavky – SQL Server | Microsoft Learn* [online]. [cit. 2022-11-22]. Dostupné z: https://learn-microsoft-com.translate.goog/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server-2019?view=sql-server-ver16&_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs
- [31] *Edice a podporované funkce SQL Server 2019 - SQL Server | Microsoft Learn* [online]. [cit. 2022-11-22]. Dostupné z: <https://learn-microsoft-com.translate.goog/en-us/sql/sql->

server/editions-and-components-of-sql-server-2019?view=sql-server-ver16&_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs

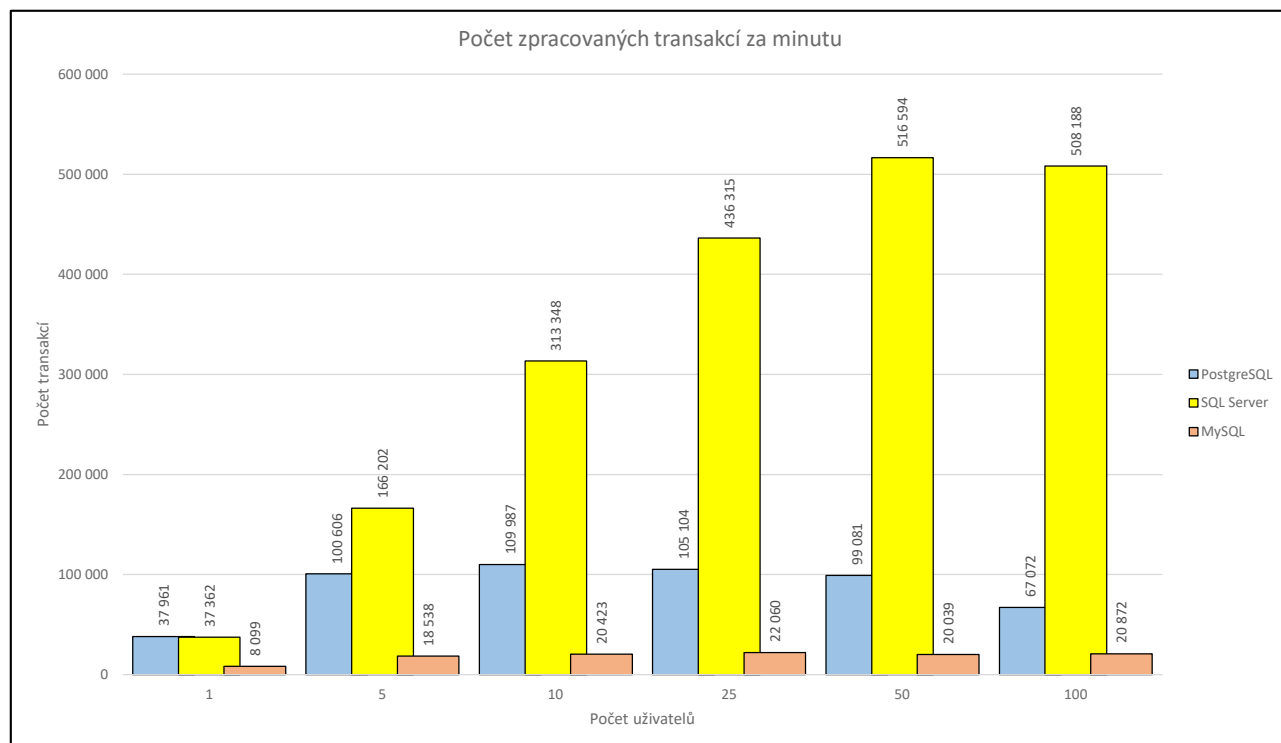
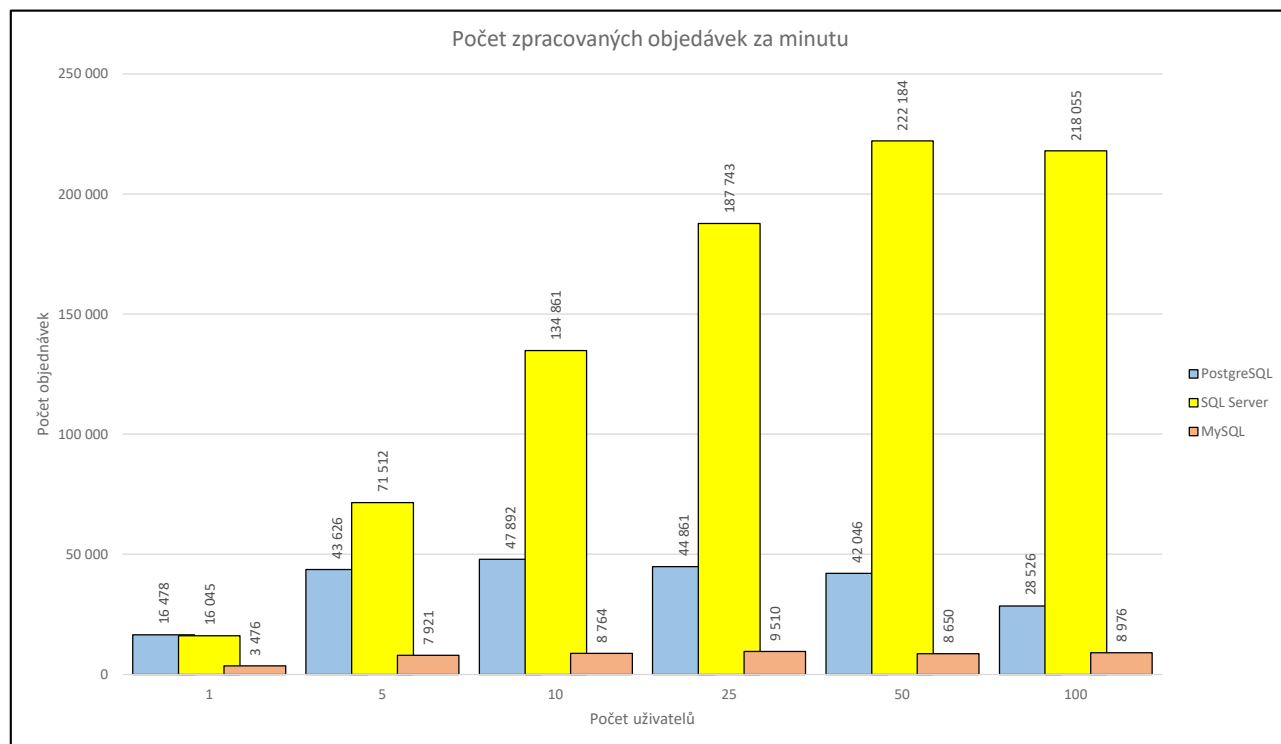
- [32] *MySQL - Wikipedia* [online]. [cit. 2023-10-06]. Dostupné z: <https://en.wikipedia.org/wiki/MySQL>
- [33] *What Is MySQL? | Oracle* [online]. [cit. 2023-09-25]. Dostupné z: <https://www.oracle.com/mysql/what-is-mysql/>
- [34] *MySQL :: MySQL Classic Edition* [online]. [cit. 2023-10-06]. Dostupné z: <https://www.mysql.com/products/classic/>
- [35] *MySQL :: MySQL Standard Edition* [online]. [cit. 2023-10-06]. Dostupné z: <https://www.mysql.com/products/standard/>
- [36] *MySQL :: MySQL Enterprise Edition* [online]. [cit. 2023-10-06]. Dostupné z: <https://www.mysql.com/products/enterprise/>
- [37] *MySQL :: MySQL Cluster CGE* [online]. [cit. 2023-10-06]. Dostupné z: <https://www.mysql.com/products/cluster/>
- [38] *MySQL :: Download MySQL Installer* [online]. [cit. 2023-10-06]. Dostupné z: <https://dev.mysql.com/downloads/installer/>
- [39] *MySQL 8 is coming | Opensource.com* [online]. [cit. 2023-10-06]. Dostupné z: <https://opensource.com/article/17/2/mysql-8-coming>
- [40] *MySQL :: MySQL Enterprise Monitor 8.0 Manual :: 3.2.1 System Requirements* [online]. [cit. 2023-10-02]. Dostupné z: <https://dev.mysql.com/doc/mysql-monitor/8.0/en/system-prereqs-reference.html>
- [41] *Microsoft SQL Server (MSSQL) vs. PostgreSQL Comparison in Details - What are the Differences? [2020]* [online]. [cit. 2023-10-11]. Dostupné z: <https://www.enterprisedb.com/blog/microsoft-sql-server-mssql-vs-postgresql-comparison-details-what-differences>
- [42] *PostgreSQL vs MySQL: Explore Their 12 Critical Differences* [online]. [cit. 2023-12-03]. Dostupné z: <https://kinsta.com/blog/postgresql-vs-mysql/>
- [43] *PostgreSQL vs. SQL Server: What's the difference? | Google Cloud* [online]. [cit. 2023-10-12]. Dostupné z: <https://cloud.google.com/learn/postgresql-vs-sql>
- [44] *SQL Data Types for MySQL, SQL Server, and MS Access* [online]. [cit. 2023-12-03]. Dostupné z: https://www.w3schools.com/sql/sql_datatypes.asp
- [45] *Geometry and Geography data types in mysql - Stack Overflow* [online]. [cit. 2023-12-06]. Dostupné z: <https://stackoverflow.com/questions/34651210/geometry-and-geography-data-types-in-mysql>
- [46] *MySQL :: MySQL 8.0 Reference Manual :: B.3.4.1 Case Sensitivity in String Searches* [online]. [cit. 2023-12-06]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/case-sensitivity.html>
- [47] *MySQL :: MySQL 8.0 Reference Manual :: 8.3.1 How MySQL Uses Indexes* [online]. [cit. 2023-12-06]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html>

- [48] *MySQL :: MySQL NDB Cluster CGE* [online]. [cit. 2023-12-07]. Dostupné z: <https://www.mysql.com/products/cluster/>
- [49] *MySQL :: MySQL Enterprise High Availability* [online]. [cit. 2023-12-07]. Dostupné z: https://www.mysql.com/products/enterprise/high_availability.html
- [50] *MySQL View - javatpoint* [online]. [cit. 2023-12-11]. Dostupné z: https://www-javatpoint-com.translate.goog/mysql-view?_x_tr_sl=auto&_x_tr_tl=cs&_x_tr_hl=cs
- [51] *MySQL :: MySQL 8.0 Reference Manual :: 25.3 Using Triggers* [online]. [cit. 2023-12-11]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/triggers.html>
- [52] *MySQL :: MySQL 8.0 Reference Manual :: 12.9 Full-Text Search Functions* [online]. [cit. 2023-12-12]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/fulltext-search.html>
- [53] *MySQL :: MySQL 8.0 Reference Manual :: 11.1.2 Integer Types (Exact Value) - INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT* [online]. [cit. 2023-12-12]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/integer-types.html>
- [54] *MySQL Data Types: Full List with Examples (2023)* [online]. [cit. 2023-12-12]. Dostupné z: <https://blog.devart.com/mysql-data-types.html>
- [55] *MySQL :: MySQL NDB Cluster: NoSQL* [online]. [cit. 2023-12-12]. Dostupné z: <https://www.mysql.com/products/cluster/nosql.html>
- [56] *Top 5 MySQL GUI Clients to Command MySQL 2023* [online]. [cit. 2023-12-13]. Dostupné z: <https://www.bytebase.com/blog/top-mysql-gui-client/>
- [57] *MySQL :: MySQL 8.0 Reference Manual :: 15.8.4 Configuring Thread Concurrency for InnoDB* [online]. [cit. 2023-12-13]. Dostupné z: https://dev.mysql.com/doc/refman/8.0/en/innodb-performance-thread_concurrency.html
- [58] *TPC-A Homepage* [online]. [cit. 2023-10-04]. Dostupné z: <https://www.tpc.org/tpca/default5.asp>
- [59] *TPC-C Homepage* [online]. [cit. 2023-09-26]. Dostupné z: <https://www.tpc.org/tpcc/>
- [60] *TPC-C Overview* [online]. [cit. 2023-10-04]. Dostupné z: <https://www.tpc.org/tpcc/detail5.asp>
- [61] *HammerDB* [online]. [cit. 2023-10-04]. Dostupné z: <https://www.hammerdb.com/about.html>
- [62] *SQL Server Benchmarking with HammerDB | Heraflux Technologies* [online]. [cit. 2023-10-31]. Dostupné z: <https://www.heraflux.com/resources/howto/sql-server-benchmarking-with-hammerdb/>
- [63] *What Is SQL Server Express and Why Would You Use It - Neovera* [online]. [cit. 2024-01-31]. Dostupné z: <https://neovera.com/sql-server-express-use/#:~:text=Limitations%20of%20SQL%20Server%20Express,SQL%20user%20connections%20NOT%20limited>

Přílohy: Výsledky jednotlivých měření a jejich grafické zpracování

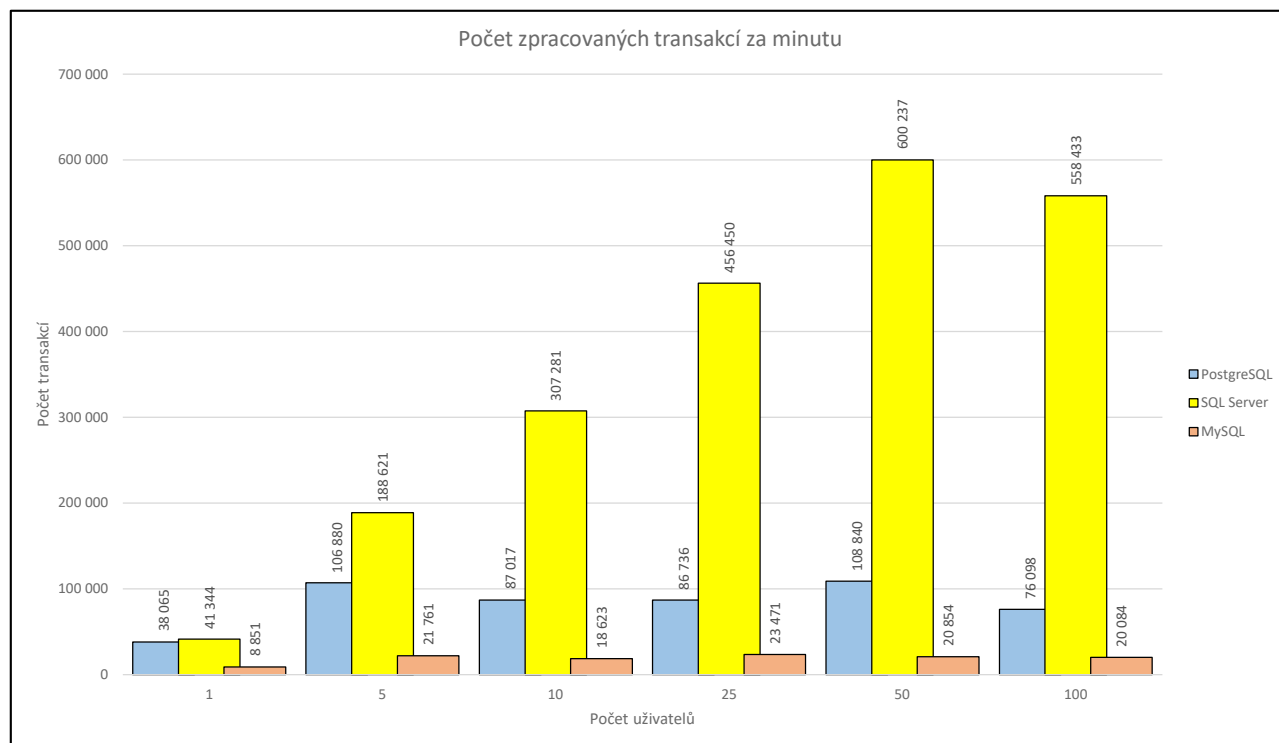
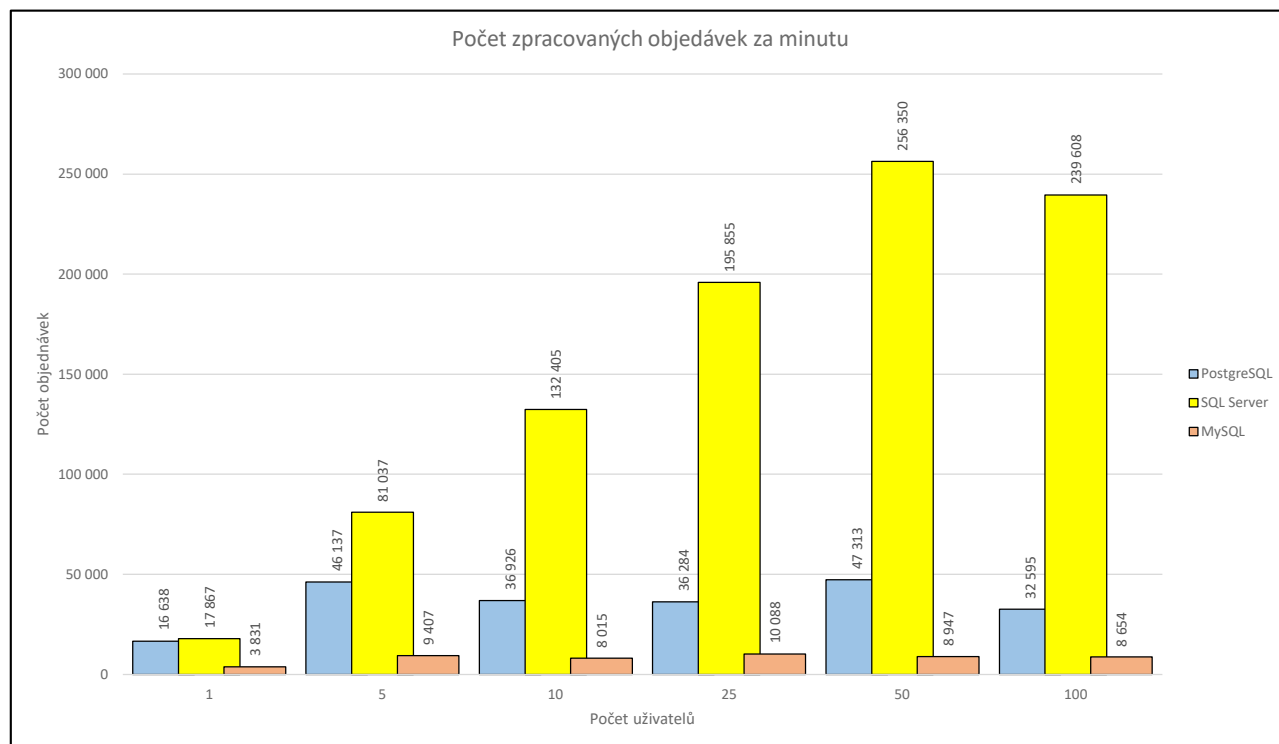
Měření 1

6.12.2023	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	13:58:11	14:08:12	14:18:16	14:28:22	14:38:36	14:49:03
		OPM	16 478	43 626	47 892	44 861	42 046	28 526
		TPM	37 961	100 606	109 987	105 104	99 081	67 072
	SQL Server	Čas	15:15:52	15:25:24	15:35:58	15:46:04	15:56:21	16:06:47
		OPM	16 045	71 512	134 861	187 743	222 184	218 055
		TPM	37 362	166 202	313 348	436 315	516 594	508 188
	MySQL	Čas	16:20:47	16:30:49	16:40:52	16:50:58	17:01:12	17:11:40
		OPM	3 476	7 921	8 764	9 510	8 650	8 976
		TPM	8 099	18 538	20 423	22 060	20 039	20 872



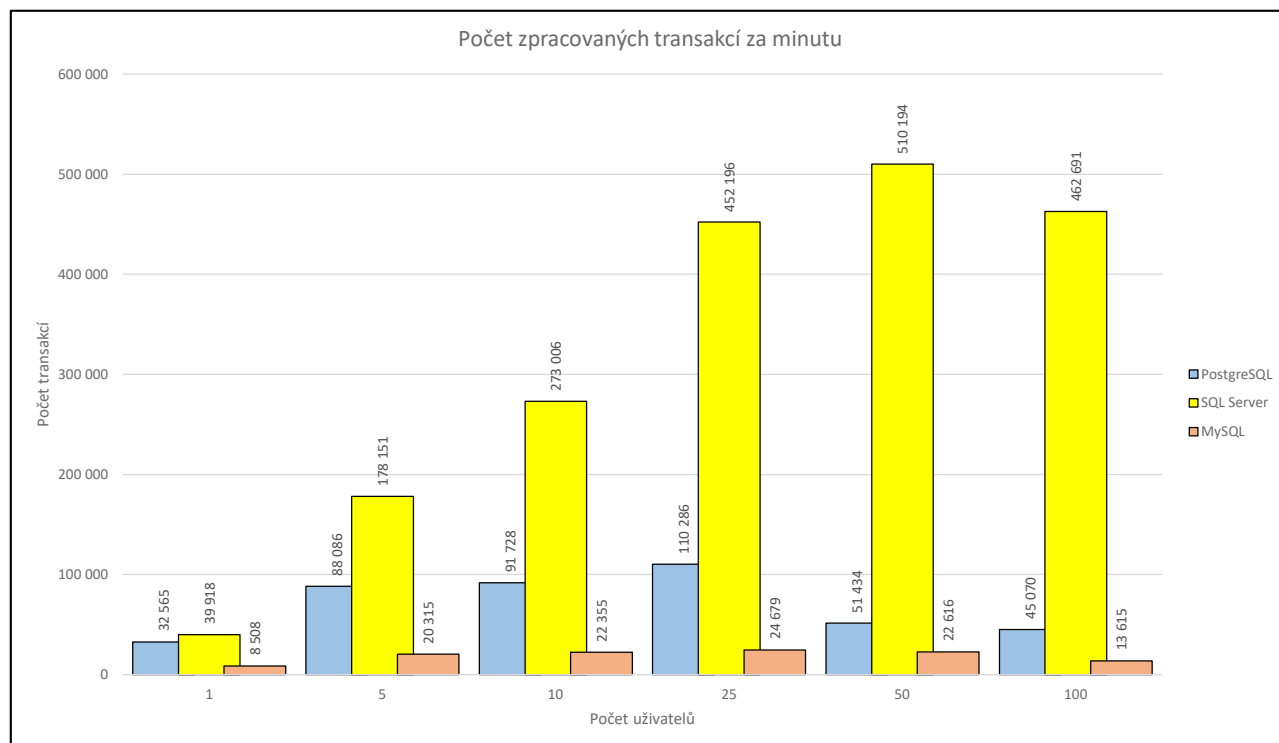
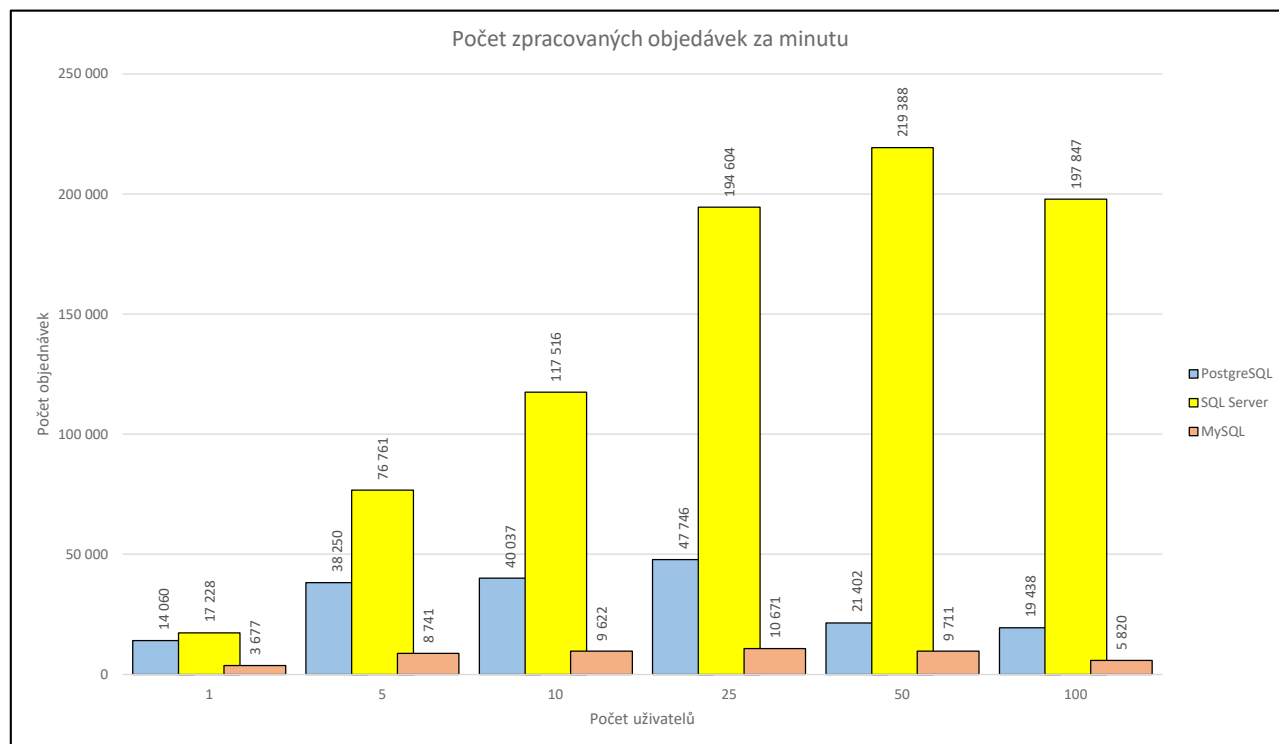
Měření 2

7.12.2023	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	11:08:42	11:18:43	11:28:48	11:38:54	11:49:09	11:59:36
		OPM	16 638	46 137	36 926	36 284	47 313	32 595
		TPM	38 065	106 880	87 017	86 736	108 840	76 098
	SQL Server	Čas	12:57:16	13:07:18	13:17:22	13:27:29	13:37:44	13:48:12
		OPM	17 867	81 037	132 405	195 855	256 350	239 608
		TPM	41 344	188 621	307 281	456 450	600 237	558 433
	MySQL	Čas	14:41:43	14:51:44	15:01:48	15:11:55	15:22:09	15:22:37
		OPM	3 831	9 407	8 015	10 088	8 947	8 654
		TPM	8 851	21 761	18 623	23 471	20 854	20 084



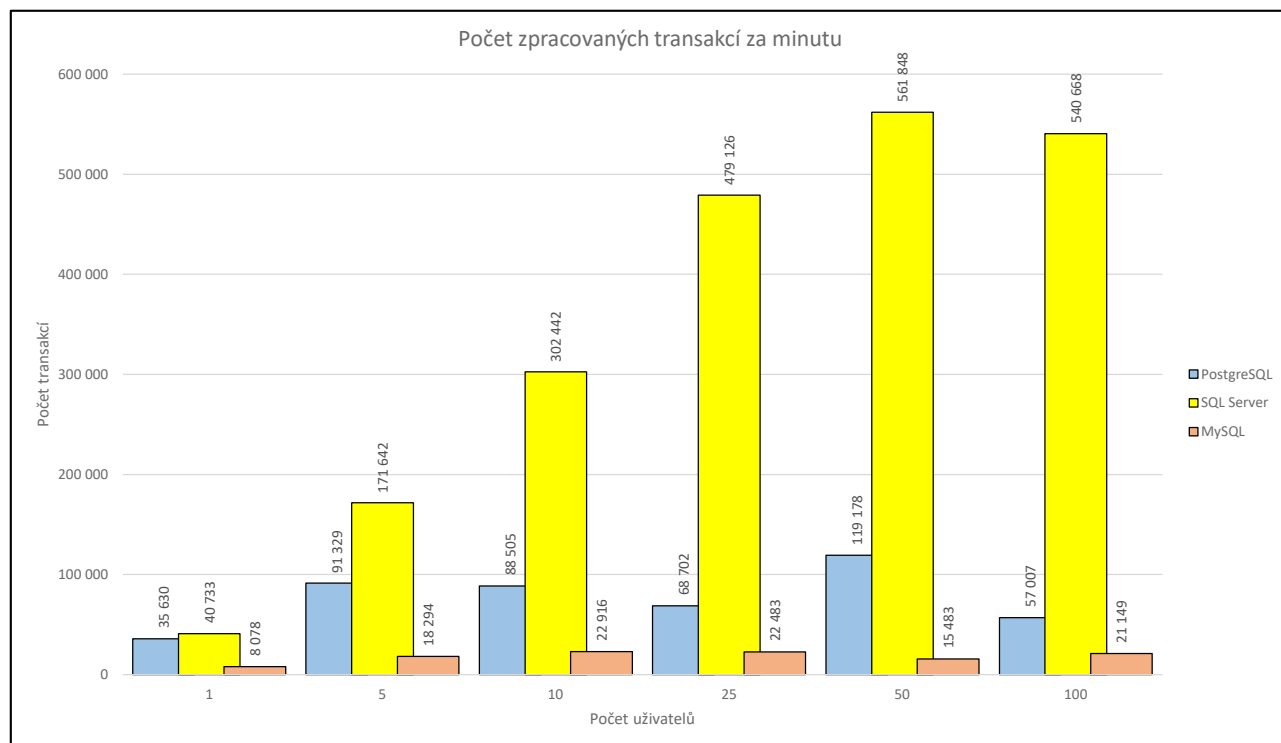
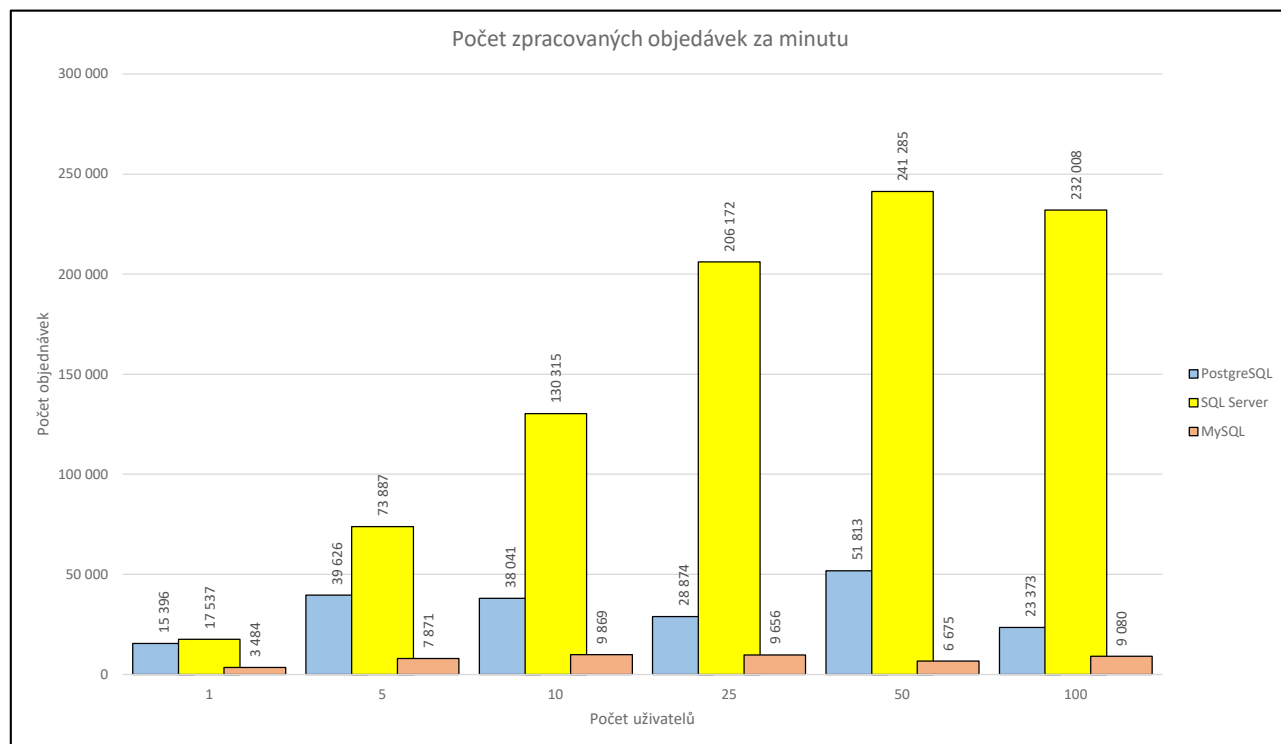
Měření 3

12.12.2023	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	9:12:28	9:22:30	9:32:35	9:42:42	9:52:56	10:03:22
		OPM	14 060	38 250	40 037	47 746	21 402	19 438
		TPM	32 565	88 086	91 728	110 286	51 434	45 070
	SQL Server	Čas	10:38:17	10:48:19	10:58:23	11:08:28	11:18:42	11:29:08
		OPM	17 228	76 761	117 516	194 604	219 388	197 847
		TPM	39 918	178 151	273 006	452 196	510 194	462 691
	MySQL	Čas	13:05:25	13:15:27	12:25:31	13:35:38	13:45:51	13:56:19
		OPM	3 677	8 741	9 622	10 671	9 711	5 820
		TPM	8 508	20 315	22 355	24 679	22 616	13 615



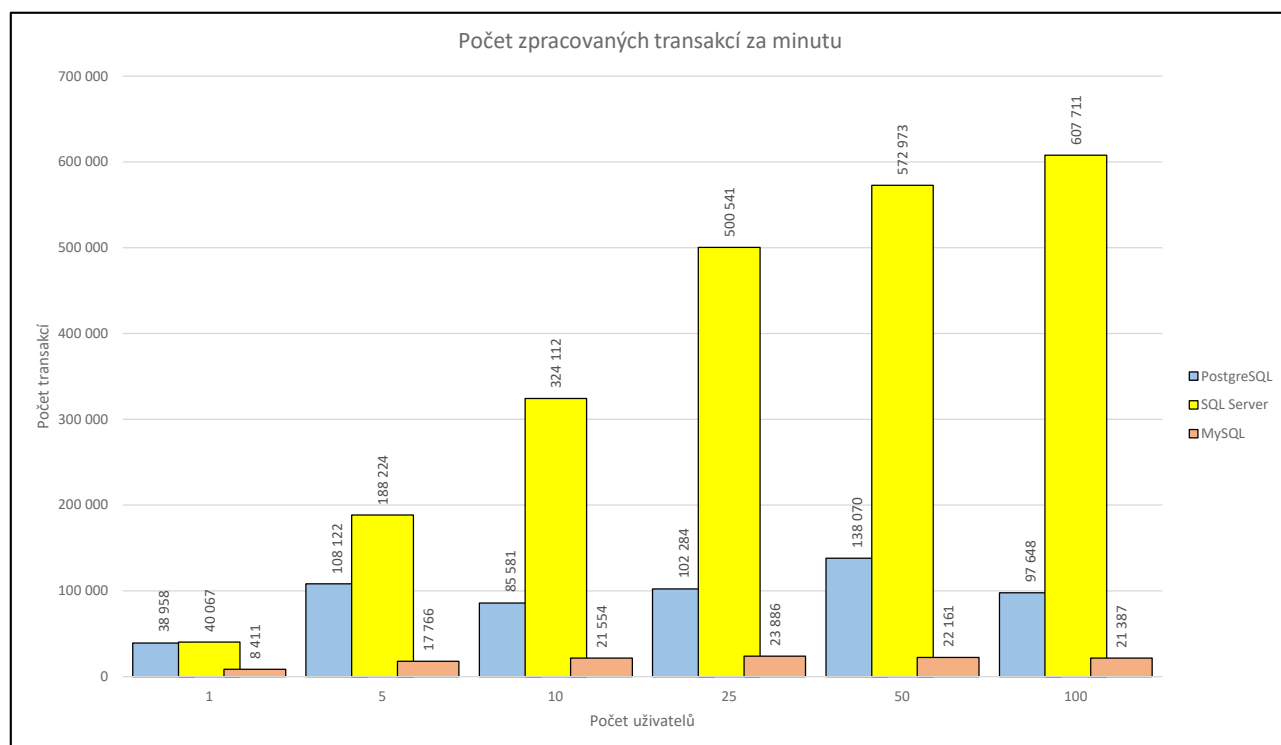
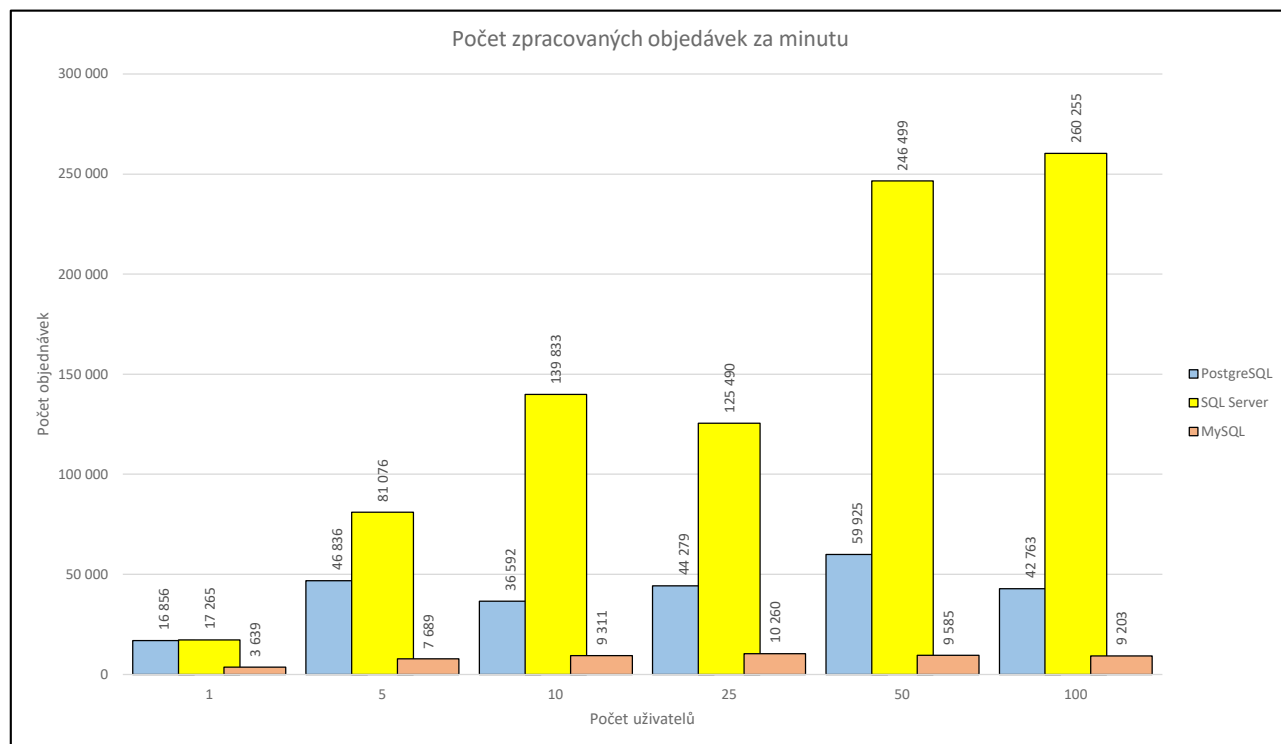
Měření 4

13.12.2023	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	9:29:57	9:39:58	9:50:02	10:00:07	10:10:21	10:20:48
		OPM	15 396	39 626	38 041	28 874	51 813	23 373
		TPM	35 630	91 329	88 505	68 702	119 178	57 007
	SQL Server	Čas	10:52:30	11:02:32	11:12:37	11:22:43	11:44:04	11:54:32
		OPM	17 537	73 887	130 315	206 172	241 285	232 008
		TPM	40 733	171 642	302 442	479 126	561 848	540 668
	MySQL	Čas	13:15:01	13:25:03	13:35:06	13:45:13	13:55:27	14:05:00
		OPM	3 484	7 871	9 869	9 656	6 675	9 080
		TPM	8 078	18 294	22 916	22 483	15 483	21 149



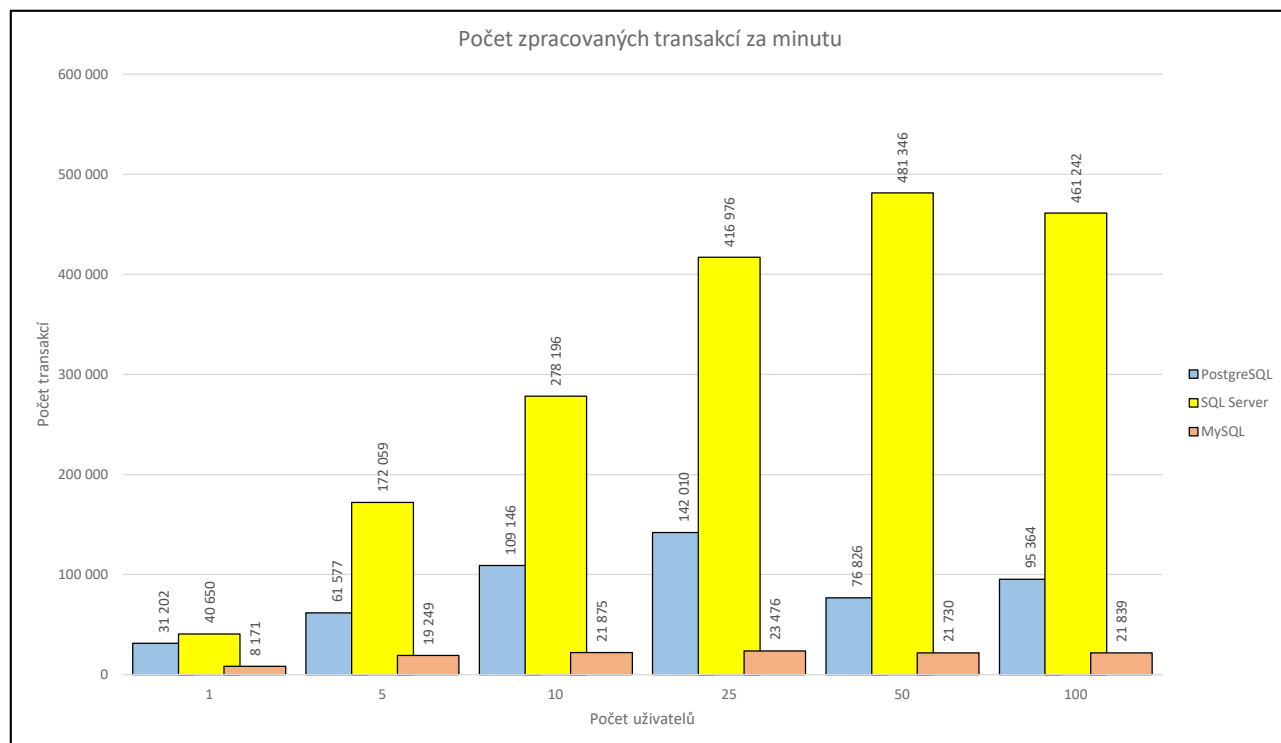
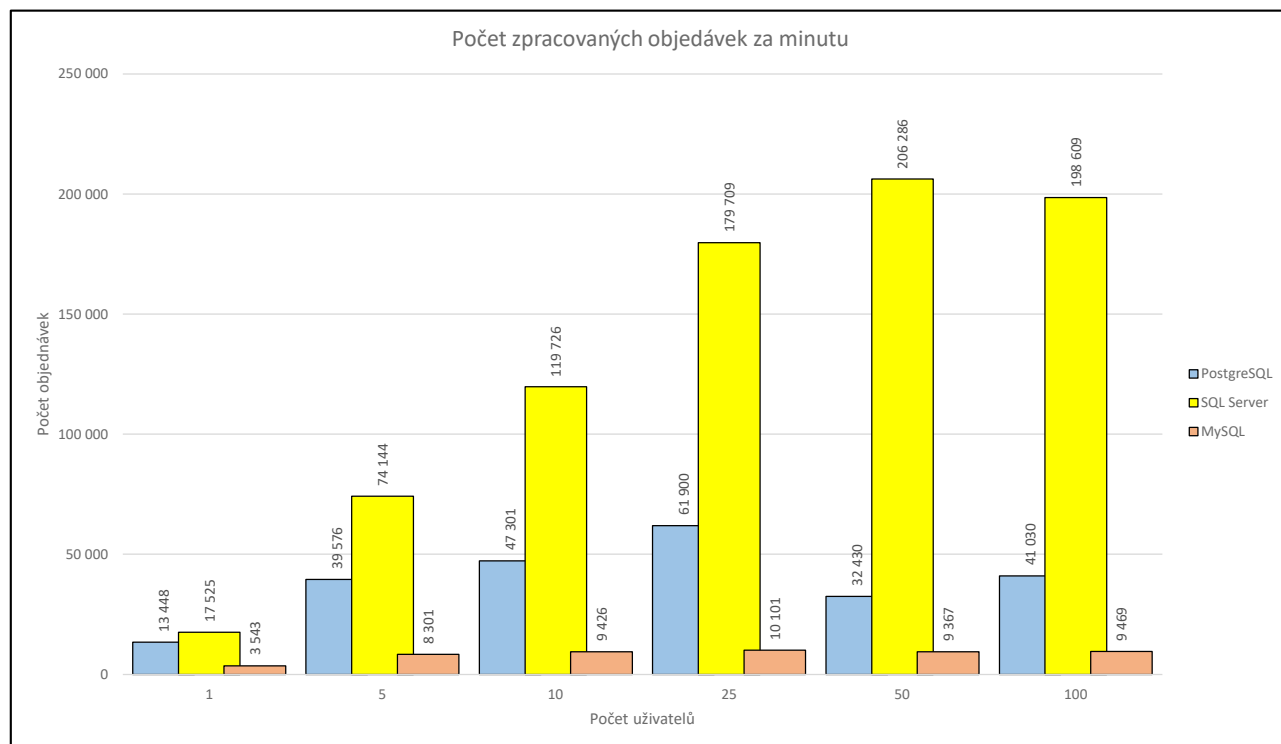
Měření 5

14.12.2023	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	9:16:59	9:27:01	9:37:05	9:47:10	9:57:24	10:07:51
		OPM	16 856	46 836	36 592	44 279	59 925	42 763
		TPM	38 958	108 122	85 581	102 284	138 070	97 648
	SQL Server	Čas	10:24:19	10:34:21	10:44:24	10:54:30	11:04:43	11:15:09
		OPM	17 265	81 076	139 833	125 490	246 499	260 255
		TPM	40 067	188 224	324 112	500 541	572 973	607 711
	MySQL	Čas	12:54:04	13:04:06	13:14:09	13:24:16	13:34:30	13:44:58
		OPM	3 639	7 689	9 311	10 260	9 585	9 203
		TPM	8 411	17 766	21 554	23 886	22 161	21 387



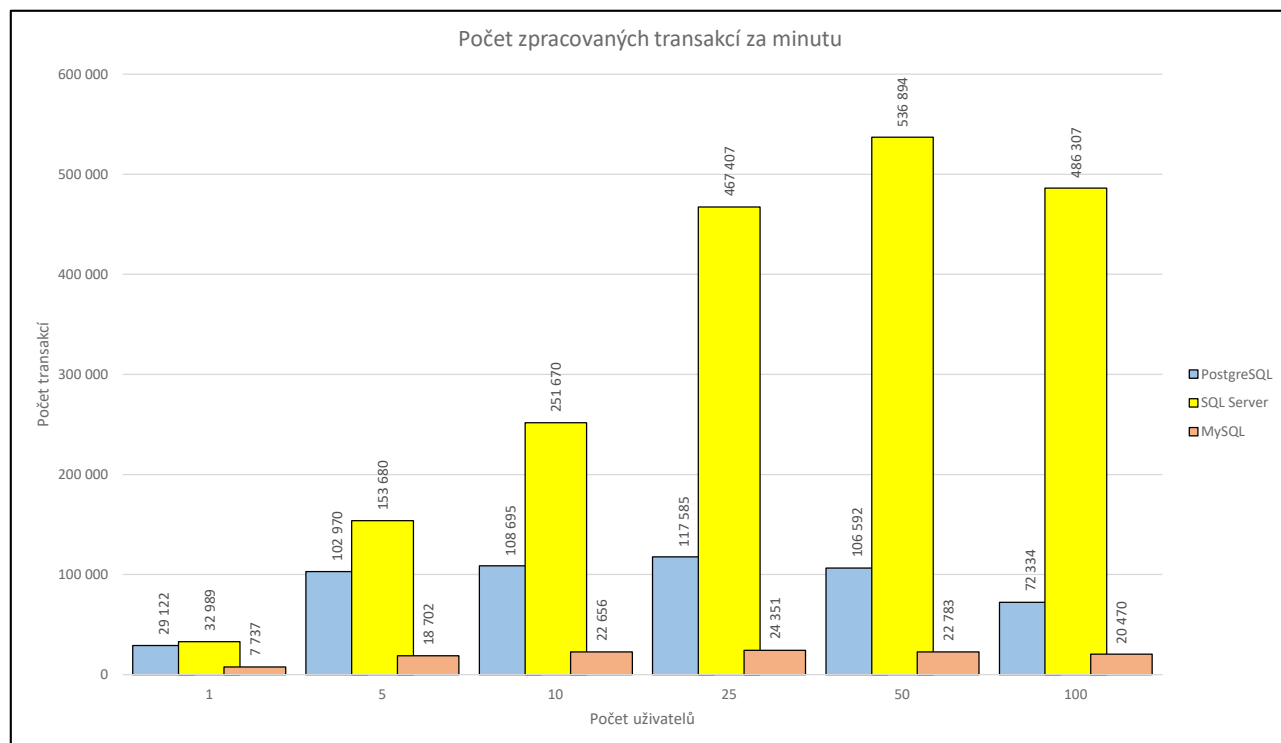
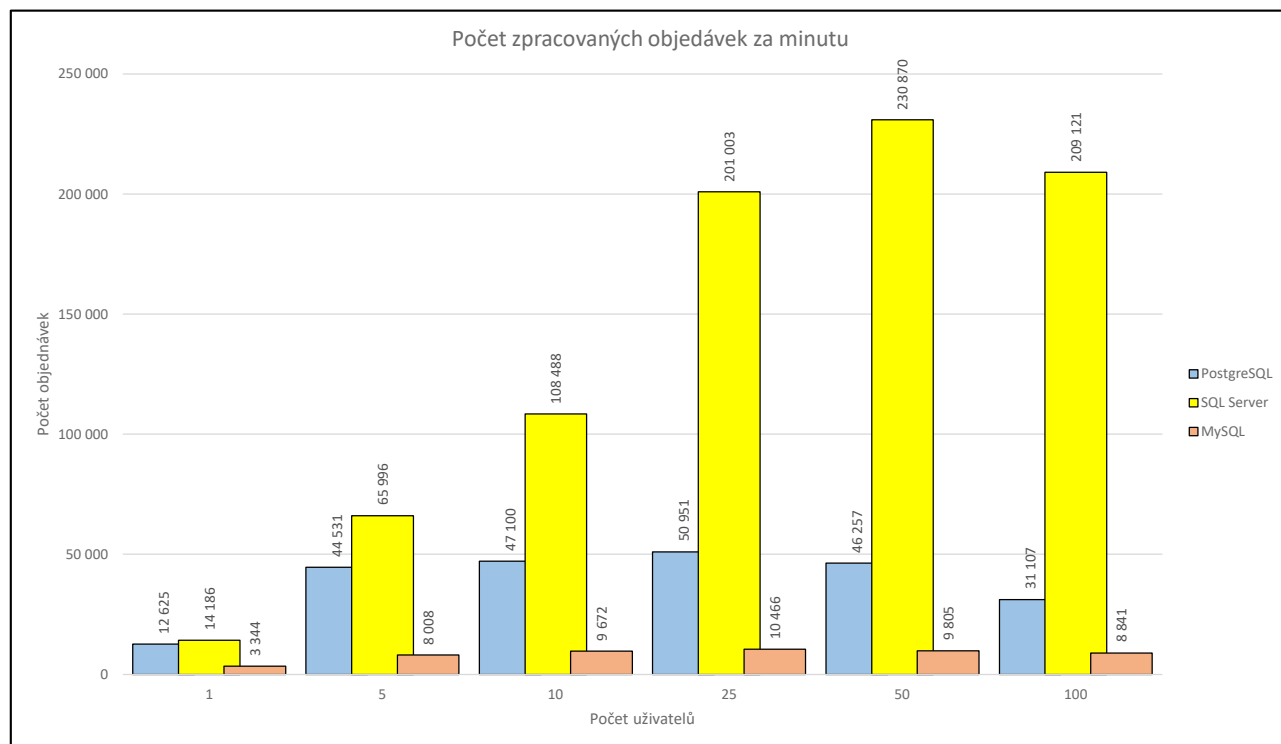
Měření 6

19.12.2023	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	11:12:58	11:23:00	11:33:04	11:43:11	11:53:26	12:03:53
		OPM	13 448	39 576	47 301	61 900	32 430	41 030
		TPM	31 202	61 577	109 146	142 010	76 826	95 364
	SQL Server	Čas	12:57:49	13:07:50	13:17:53	13:27:59	14:43:54	14:54:22
		OPM	17 525	74 144	119 726	179 709	206 286	198 609
		TPM	40 650	172 059	278 196	416 976	481 346	461 242
	MySQL	Čas	15:34:18	15:44:20	15:54:24	16:04:31	16:14:45	16:25:12
		OPM	3 543	8 301	9 426	10 101	9 367	9 469
		TPM	8 171	19 249	21 875	23 476	21 730	21 839



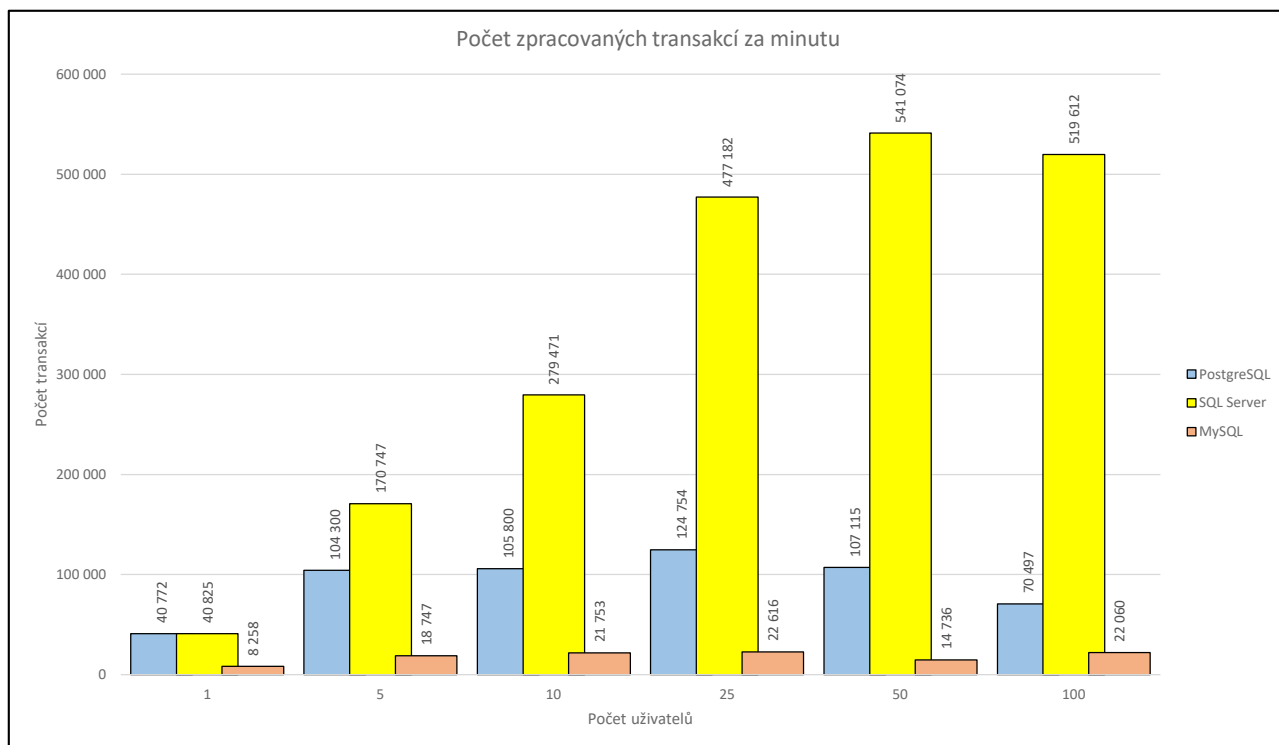
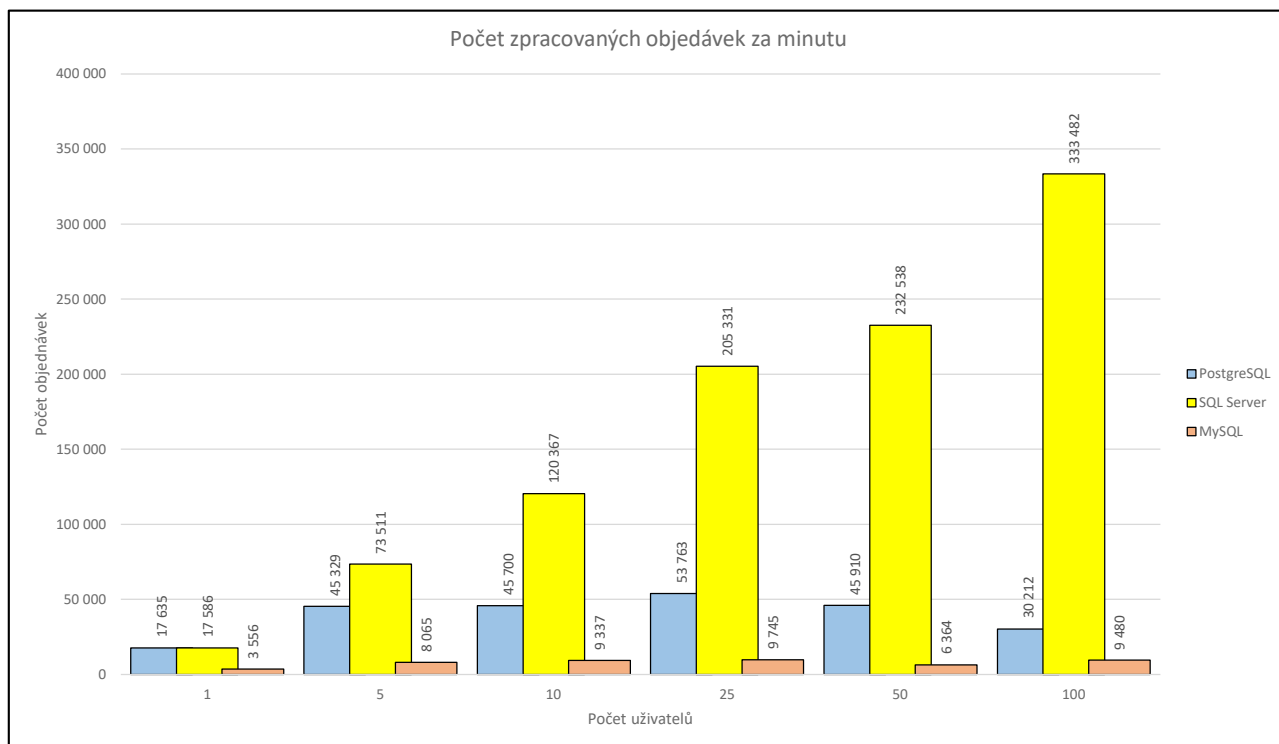
Měření 7

2.1.2024	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	9:49:44	9:59:46	10:09:50	10:19:55	10:30:11	10:40:39
		OPM	12 625	44 531	47 100	50 951	46 257	31 107
		TPM	29 122	102 970	108 695	117 585	106 592	72 334
	SQL Server	Čas	12:40:55	12:50:57	13:01:00	13:11:07	13:21:23	13:31:51
		OPM	14 186	65 996	108 488	201 003	230 870	209 121
		TPM	32 989	153 680	251 670	467 407	536 894	486 307
	MySQL	Čas	15:11:44	15:21:45	15:31:49	15:41:56	15:52:11	16:02:39
		OPM	3 344	8 008	9 672	10 466	9 805	8 841
		TPM	7 737	18 702	22 656	24 351	22 783	20 470



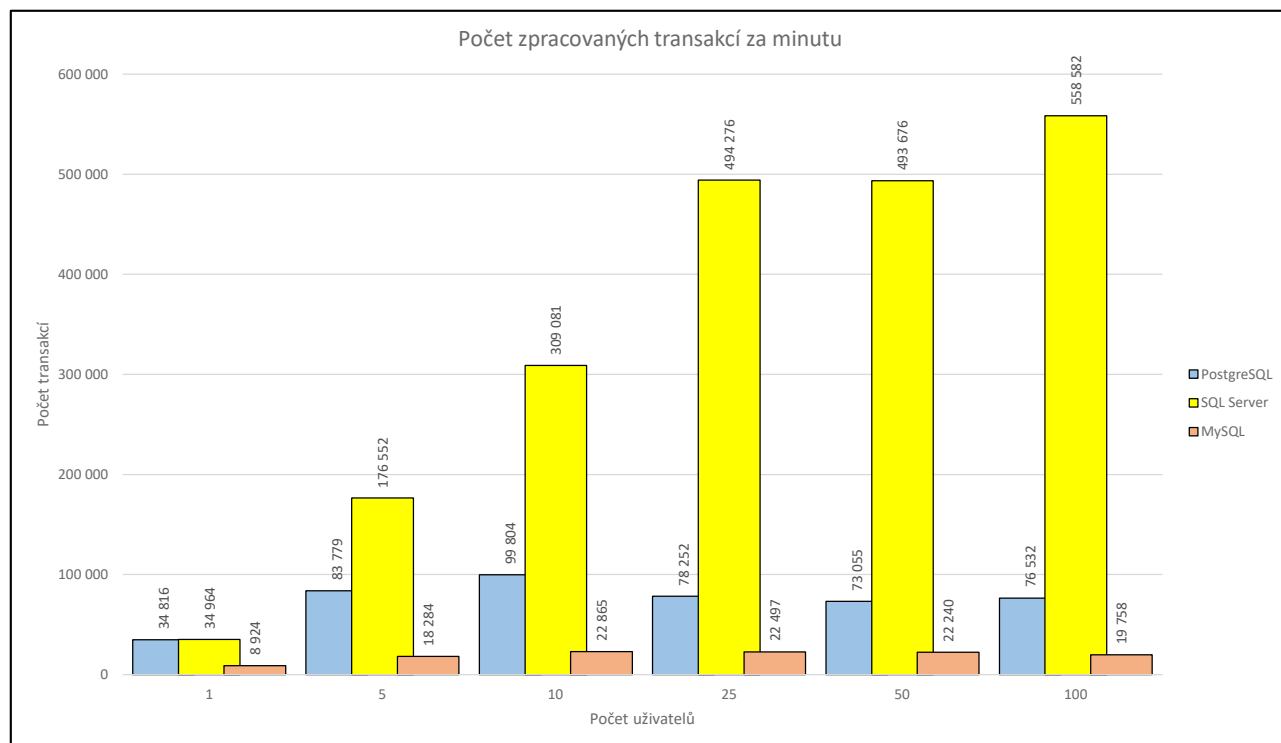
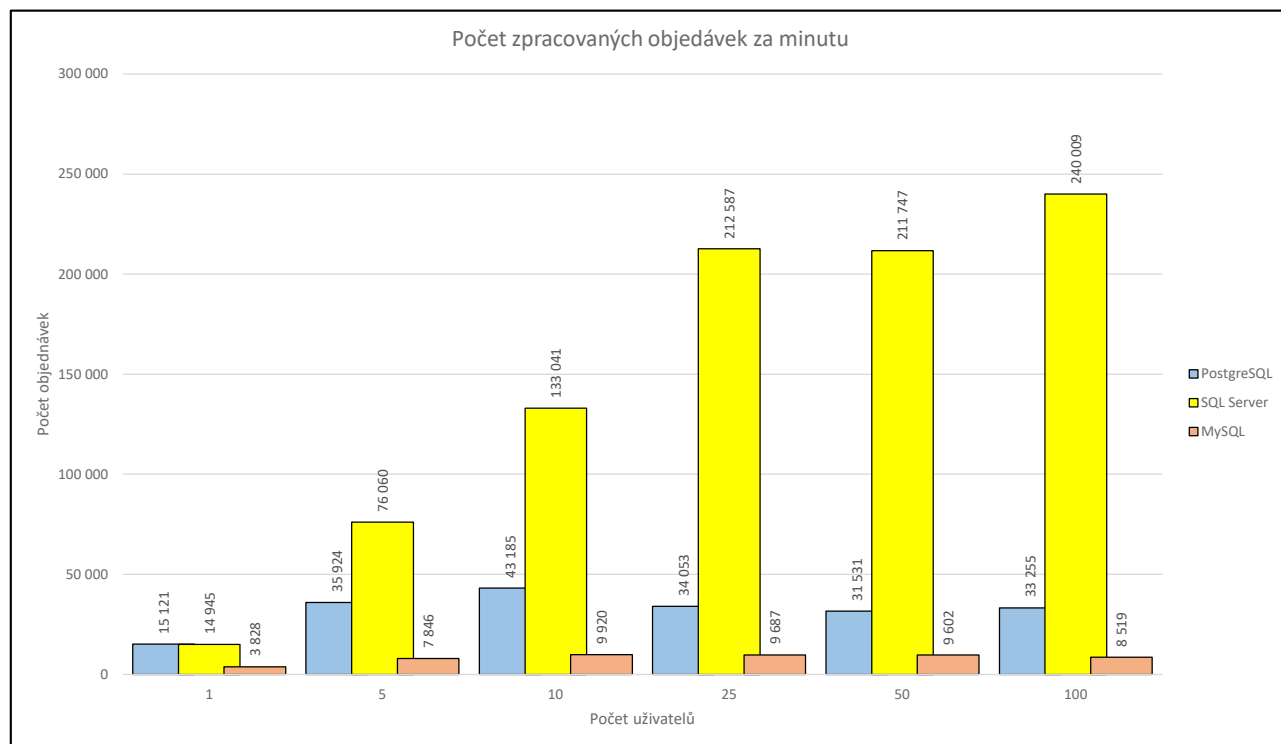
Měření 8

3.1.2024	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	10:00:58	10:11:00	10:21:03	10:31:09	10:41:23	10:51:50
		OPM	17 635	45 329	45 700	53 763	45 910	30 212
		TPM	40 772	104 300	105 800	124 754	107 115	70 497
	SQL Server	Čas	12:14:36	12:27:38	12:37:42	12:47:50	12:58:04	13:08:33
		OPM	17 586	73 511	120 367	205 331	232 538	333 482
		TPM	40 825	170 747	279 471	477 182	541 074	519 612
	MySQL	Čas	14:18:05	14:28:07	14:38:11	14:48:18	14:58:33	15:09:00
		OPM	3 556	8 065	9 337	9 745	6 364	9 480
		TPM	8 258	18 747	21 753	22 616	14 736	22 060



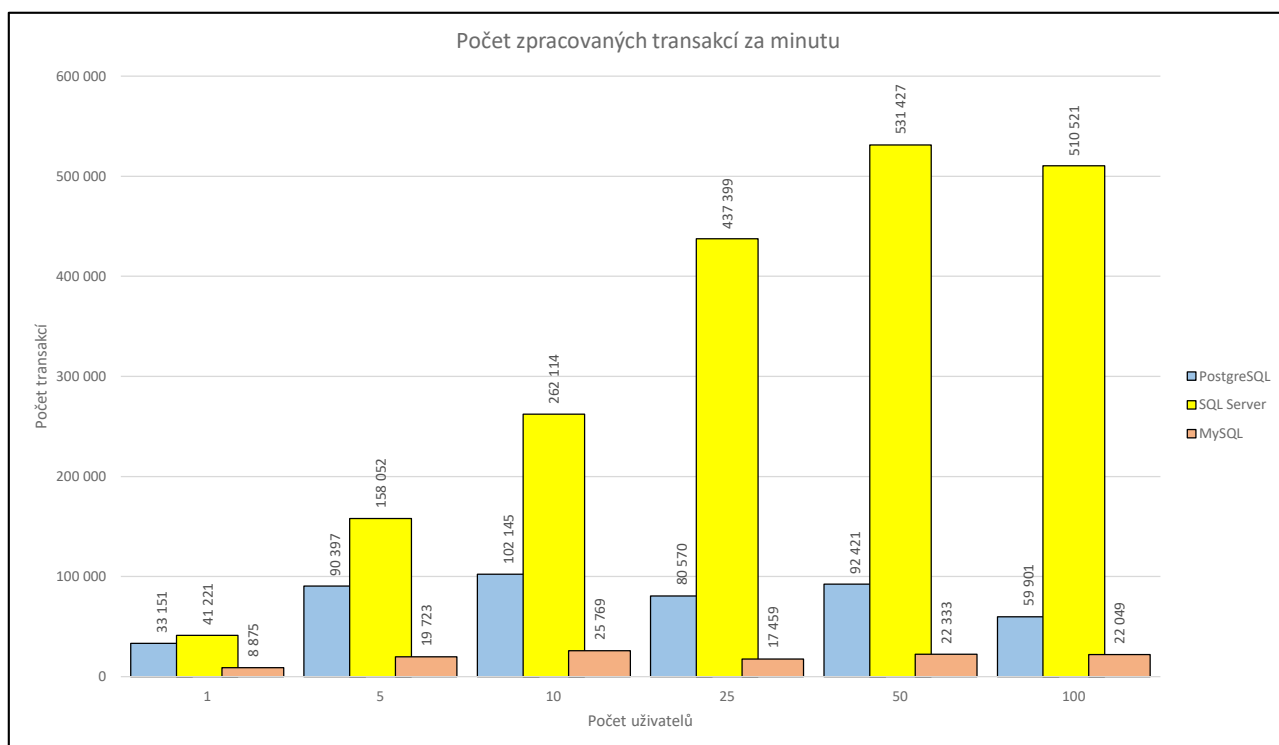
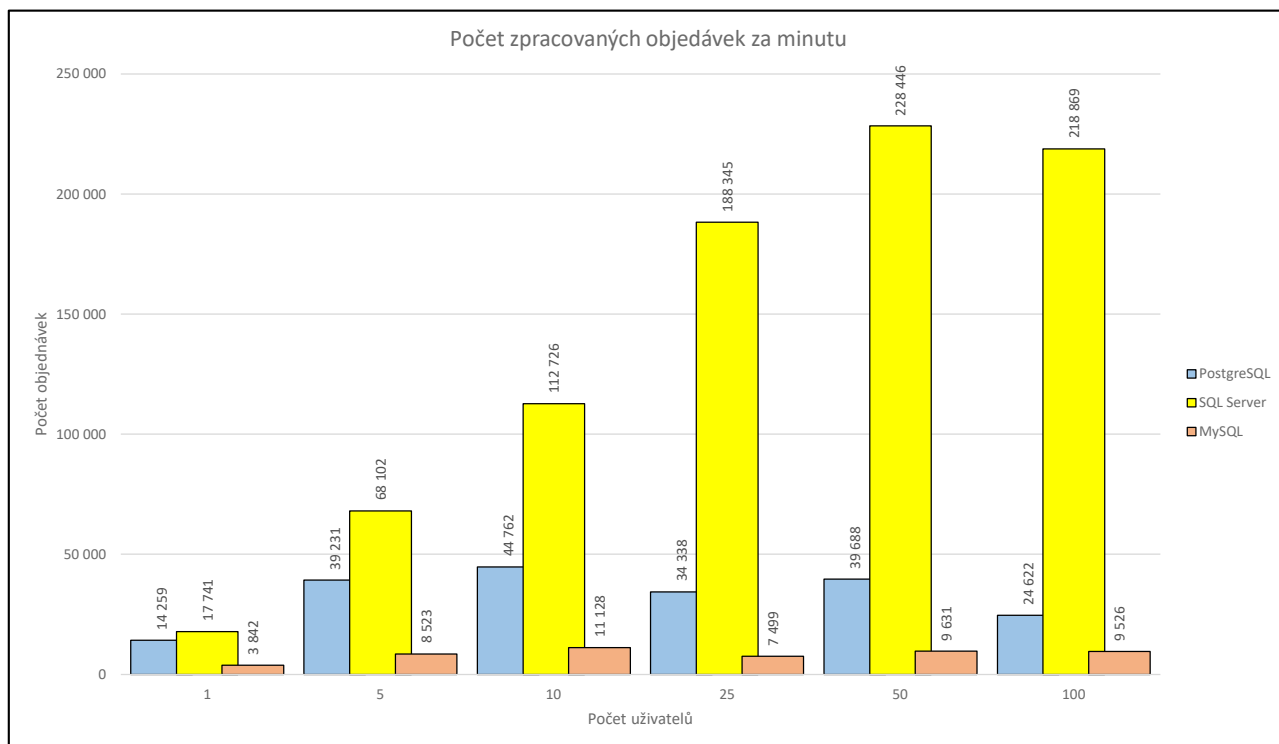
Měření 9

5.1.2024	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	8:59:27	9:09:30	9:19:34	9:29:40	9:39:54	9:50:21
		OPM	15 121	35 924	43 185	34 053	31 531	33 255
		TPM	34 816	83 779	99 804	78 252	73 055	76 532
	SQL Server	Čas	10:15:05	10:25:08	10:35:12	10:45:19	10:55:32	11:05:59
		OPM	14 945	76 060	133 041	212 587	211 747	240 009
		TPM	34 964	176 552	309 081	494 276	493 676	558 582
	MySQL	Čas	12:11:36	12:21:37	12:31:42	12:41:49	12:52:04	13:02:32
		OPM	3 828	7 846	9 920	9 687	9 602	8 519
		TPM	8 924	18 284	22 865	22 497	22 240	19 758



Měření 10

8.1.2024	Počet uživatelů		1	5	10	25	50	100
	PostgreSQL	Čas	8:57:41	9:07:43	9:17:47	9:27:54	9:38:08	9:48:35
		OPM	14 259	39 231	44 762	34 338	39 688	24 622
		TPM	33 151	90 397	102 145	80 570	92 421	59 901
	SQL Server	Čas	10:29:37	10:39:39	10:49:43	10:59:50	11:10:04	11:20:32
		OPM	17 741	68 102	112 726	188 345	228 446	218 869
		TPM	41 221	158 052	262 114	437 399	531 427	510 521
	MySQL	Čas	12:29:24	12:39:25	12:49:29	12:59:35	13:09:51	12:20:18
		OPM	3 842	8 523	11 128	7 499	9 631	9 526
		TPM	8 875	19 723	25 769	17 459	22 333	22 049



Měření SQL Express

		Počet uživatelů	1	5	10	25	50	100
9.1.2024	1	Čas	11:30:02	11:40:03	11:50:07	12:00:13	12:10:28	12:20:54
		OPM	15 472	61 356	91 942	97 574	98 275	97 069
		TPM	35 718	142 677	213 215	226 201	228 097	225 790
	2	Čas	12:42:38	12:52:40	13:02:44	13:12:51	13:23:05	13:33:31
		OPM	15 483	65 462	85 899	99 133	101 339	101 544
		TPM	35 958	152 182	199 935	230 612	235 512	235 919
	3	Čas	14:57:39	15:07:41	15:17:45	15:27:51	15:38:06	15:48:33
		OPM	15 375	60 996	88 278	100 174	100 034	93 635
		TPM	35 709	141 596	204 863	233 201	232 303	217 768
10.1.2024	4	Čas	9:36:01	9:46:02	9:56:05	10:06:11	10:16:25	10:26:52
		OPM	14 245	59 102	91 389	97 875	99 598	96 381
		TPM	33 217	137 244	212 395	227 618	231 626	223 767
	5	Čas	10:45:17	10:55:18	11:05:23	11:15:30	11:25:44	11:36:11
		OPM	15 945	53 049	86 021	102 795	99 219	95 116
		TPM	36 961	123 493	200 189	239 305	230 938	220 901
	6	Čas	12:42:22	12:52:24	13:02:29	13:12:36	13:22:51	13:33:21
		OPM	16 399	64 717	83 179	103 978	101 894	93 936
		TPM	38 079	150 465	192 754	241 581	236 660	218 826
11.1.2024	7	Čas	10:28:00	10:38:01	10:48:04	10:58:10	11:08:24	11:18:51
		OPM	15 256	66 946	89 127	104 170	97 893	95 388
		TPM	35 509	155 689	207 246	241 778	227 519	222 446
	8	Čas	13:19:04	13:29:06	13:39:09	13:49:15	13:59:28	14:09:55
		OPM	15 759	60 523	88 636	100 907	87 791	95 846
		TPM	36 654	140 789	205 675	234 605	227 023	215 901
	9	Čas	14:34:35	14:44:37	14:54:40	15:04:46	15:15:00	15:25:26
		OPM	15 527	63 904	86 633	95 200	96 517	95 927
		TPM	36 110	148 618	201 226	221 066	224 232	223 447
12.1.2024	10	Čas	10:10:26	10:20:27	10:30:31	10:40:36	10:50:49	11:01:16
		OPM	15 952	64 465	87 761	102 140	100 614	90 441
		TPM	37 044	149 809	203 850	237 123	233 667	209 079
		Čas						
		OPM						
		TPM						
		Čas						
		OPM						
		TPM						