

Vysoká škola logistiky o.p.s.

**Tecnomatix Plant Simulation - využití
databází a programování v SimTalk**

(Bakalářská práce)



**Vysoká škola
logistiky**
o.p.s.

Zadání bakalářské práce

student	Martin Chytka
studijní program specializace	LOGISTIKA Informatika pro logistiku

Vedoucí Katedry bakalářského studia Vám ve smyslu čl. 22 Studijního a zkušebního řádu Vysoké školy logistiky o.p.s. pro studium v bakalářském studijním programu určuje tuto bakalářskou práci:

Název tématu: **Tecnomatix Plant Simulation - využití databází a programování v SimTalk**

Cíl práce:

Ověřit možnosti připojení Tecnomatix Plant Simulation k různým databázím a navrhnout případové studie využití databází a programování v SimTalk k automatizaci rutinních činností v Plant Simulation.

Zásady pro vypracování:

Využijte teoretických východisek oboru logistika. Čerpejte z literatury doporučené vedoucím práce a při zpracování práce postupujte v souladu s pokyny VŠLG a doporučeními vedoucího práce. Části práce využívající neveřejné informace uveďte v samostatné příloze.

Bakalářskou práci zpracujte v těchto bodech:

Úvod

1. Teoretická východiska počítačové simulace
2. Analýza databází vhodných k použití s Plant Simulation
3. Programování v SimTalk pro využití databází v Plant Simulation
4. Ukázkové příklady

Závěr

Rozsah práce: 35 – 50 normostran textu

Seznam odborné literatury:

BANGSOW, Steffen: Manufacturing Simulation with Plant Simulation and SimTalk: Usage and Programming with Examples and Solutions. Springer-Verlag Berlin Heidelberg. 2010. 297 s. ISBN 978-3-642-05073-2.

BANGSOW, Steffen. Tecnomatix Plant Simulation: Modeling and Programming by Means of Examples - Second Edition. Springer Nature Switzerland AG, 2020. ISBN 978-3-030-41544-0.

TREBUŇA, Peter - MARKOVIČ, Jaromír - KLIMENT, Marek - HALČÍNOVÁ, Jana: Modelovanie v priemyselnom inžinierstve. Prvé vydanie. Košice: TU v Košiciach, Strojnícka fakulta, 2015. 195 s. Edícia vedeckej a odbornej literatúry. ISBN 978-80-553-1953-7.

Vedoucí bakalářské práce:

prof. Ing. Gabriel Fedorko, PhD.

Datum zadání bakalářské práce:

31. 10. 2022

Datum odevzdání bakalářské práce:

29. 4. 2023

Přerov 31. 10. 2022



Ing. et Ing. Iveta Dočkalíková, Ph.D.
vedoucí katedry

Ko



prof. Ing. Václav Cempírek, Ph.D.
rektor

Čestné prohlášení

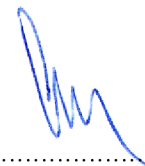
Prohlašuji, že předložená bakalářská práce je původní, a že jsem ji vypracoval samostatně. Prohlašuji, že citace použitých pramenů je úplná, a že jsem v práci neporušil autorská práva ve smyslu zákona č. 121/2000 Sb.; o autorském právu, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

Prohlašuji, že jsem byl také seznámen s tím, že se na mou bakalářskou práci plně vztahuje zákon č. 121/2000 Sb., o právu autorském, právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména § 60 – školní dílo. Beru na vědomí, že Vysoká škola logistiky o.p.s. nezasahuje do mých autorských práv užitím mé bakalářské práce pro pedagogické, vědecké a prezentační účely školy. Užiji-li svou bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat předtím o této skutečnosti prorektora pro vzdělávání Vysoké školy logistiky o.p.s.

Prohlašuji, že jsem byl poučen o tom, že bakalářská práce je veřejná ve smyslu zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, zejména § 47b. Taktéž dávám souhlas Vysoké škole logistiky o.p.s. ke zpřístupnění mnou zpracované bakalářské práce v její tištěné i elektronické verzi. Souhlasím s případným použitím této práce Vysokou školou logistiky o.p.s. pro pedagogické, vědecké a prezentační účely.

Prohlašuji, že odevzdaná tištěná verze bakalářské práce a verze nahraná do informačního systému školy jsou totožné.

V Přerově, dne 29.4. 2023



.....
podpis

Poděkování

Děkuji vedoucímu bakalářské práce prof. Ing. Gabrielu Fedorkovi, Ph.D. za ochotu, cenné rady a za pomoc při vedení této práce.

Anotace

Současný trend přechodu na digitální a automatizovanou výrobu, obecně označován jako Průmysl 4.0, vyžaduje rozsáhlé změny v přístupu k návrhu produktů, technologií, výrobních linek a mnoha dalších záležitostí s takovou výrobou souvisejících. Tlak na zkrácení času mezi návrhem výrobku, prototypem a výrobou mj. zvyšuje požadavky na automatizaci opakovaných rutinních postupů při práci obecně s každým softwarem, kde to bude možné a vhodné. Cílem této bakalářské práce je ověřit možnosti připojení simulačního software Tecnomatix Plant Simulation k různým databázím a navrhnout případové studie využití databází a programování v SimTalk k automatizaci rutinních činností v Plant Simulation.

Klíčová slova

Logistika, výroba, simulace procesů, Tecnomatix Plant Simulation, databáze, SimTalk

Annotation

The current trend of transition to digital and automated production, generally referred to as Industry 4.0, requires extensive changes in the approach to product design, technologies, production lines and many other matters related to such production. The pressure to shorten the time between product design, prototype and production, among other things, increases the demand for automation of repetitive routine procedures when working in general with any software, where possible and appropriate. The goal of this bachelor's thesis is to verify the possibilities of connecting the simulation software Tecnomatix Plant Simulation to various databases and to propose case studies of the use of databases and programming in SimTalk to automate routine activities in Plant Simulation.

Keywords

Logistics, production, process simulation, Tecnomatix Plant Simulation, databases, SimTalk

Obsah

Úvod.....	10
1 Teoretická východiska počítačové simulace	11
1.1 Obecné principy simulace	11
1.1.1 Formulace problémů	12
1.1.2 Test způsobilosti simulace	13
1.1.3 Formulace cílů	13
1.1.4 Sběr dat	13
1.1.5 Modelování	14
1.1.6 Provedení simulace	14
1.1.7 Analýza výsledků a jejich interpretace	15
1.1.8 Dokumentace	15
1.2 Proces počítačového modelování	15
1.3 Vhodné případy pro využití počítačové simulace	17
1.4 Výhody a nevýhody počítačové simulace	18
2 Analýza databází vhodných k použití s Plant Simulation	19
2.1 Tecnomatix Plant Simulation	21
2.2 DBMS – nástroje na správu relačních databází	22
2.1.1 Microsoft Office Access	22
2.1.2 LibreOffice / Apache OpenOffice	23
2.1.3 HeidiSQL	25
2.1.4 DB Browser for SQLite (DB4S).....	25
2.1.5 phpMyAdmin.....	26
2.1.6 Microsoft SQL Server Management Studio	27
2.1.7 Oracle SQL Developer.....	29
2.3 Relační databáze pro použití s Plant Simulation.....	31
2.2.1 SQLite - interní implementace v Plant Simulation.....	31

2.2.2	MSO Access	31
2.2.3	MySQL / MariaDB	31
2.2.4	Microsoft SQL Server Express	32
2.2.5	Oracle Database	32
2.4	Shrnutí analýzy databází vhodných pro Plant Simulation	33
3	Programování v SimTalk pro využití databází v Plant Simulation	34
3.1	SQLite	34
3.2	Open Database Connectivity	35
3.2.1	Nastavení připojení MySQL / MariaDB přes ODBC	36
3.2.2	Nastavení připojení MS SQL Server Express přes ODBC	37
3.3	Oracle11g	39
3.4	Oracle19c	39
4	Ukázkové příklady	40
4.1	Plant Simulation, SimTalk a interní DB SQLite	43
4.1.1	Programový kód pro vytvoření tabulek, metoda <i>Model.Models.Init</i>	43
4.1.2	Programový kód pro zápis proměnných <i>Iter01</i> a <i>Iter02</i>	44
4.1.3	Zobrazení databáze <i>tmx_sqlite.db</i> v DB Browser for SQLite	44
4.2	Plant Simulation, SimTalk a připojení přes ODBC k externí síťové databázi MariaBD	46
4.2.1	Programový kód pro vytvoření tabulek, metoda <i>Model.Models.Init</i>	46
4.2.2	Programový kód pro zápis proměnných <i>Iter01</i> a <i>Iter02</i>	47
4.3	Plant Simulation, SimTalk a připojení přes ODBC k externí lokální databázi MS SQL Server Express	50
4.3.1	Programový kód pro vytvoření tabulek, metoda <i>Model.Models.Init</i>	51
4.3.2	Programový kód pro zápis proměnných <i>Iter01</i> a <i>Iter02</i>	52
	Závěr	55
	Seznam zdrojů	56
	Seznam grafických objektů	58

Seznam zkratek	59
----------------------	----

Úvod

V dnešní době existují velké balíky softwarových řešení pro průmysl, které nabízejí velmi rozsáhlé obecné možnosti použití. Klientské přizpůsobování těchto řešení konkrétním potřebám zákazníků může umožňovat zrychlení práce s takovýmto software. Výrobci software dnes v mnoha případech jako standard nabízejí v rámci svého software rozhraní API, které umožňuje rozsáhlé uživatelské přizpůsobení tohoto software doprogramováním potřebných doplňků, nebo rozšíření programovým doplňkem jak třetí strany, tak i svépomocí zákazníka, pokud disponuje přiměřeně zkušeným IT oddělením. Mnoho softwarových produktů tak lze programově uživatelsky rozšiřovat o potřebnou funkčnost, kterou produkt nenabízí jako standardní součást.

Některé jiné produkty sice komplexní API nenabízejí, ale nabízejí např. interní programovací, nebo skriptovací jazyk pro možnost uživatelské individualizace software.

Samostatnou kapitolou je pak napojení jednoho software na nějaký jiný software za účelem např. přenosu dat z jednoho SW do jiného a zpracování těchto dat v jiném SW, než ve kterém data vznikla. Typickým příkladem této situace může být SW pro simulaci výrobních a logistických procesů a jeho napojení na externí databáze, přenos dat ze simulačního SW do databází a zpracovávání, nebo ukládání těchto dat v databázích. Tomuto případu se také věnuje tato práce.

1 Teoretická východiska počítačové simulace

Vlivem mezinárodní konkurence výrazně roste tlak na zvyšování efektivity výrobních systémů. Také se zvyšuje počet součástí v produktech a v důsledku toho rostou i požadavky na montážní procesy a logistické procesy. Mezinárodní logistické sítě vyžadují odpovídající logistické koncepce. Tyto požadavky lze spravovat pouze pomocí vhodných nástrojů tzv. digitální továrny v kontextu prostředí pro správu životního cyklu produktu, které umožňuje opakované použití dat, podporuje efektivní spolupráci mezi různými odděleními a poskytuje aktuální a relevantní data každému uživateli, který to potřebuje. Simulace kompletního materiálového toku včetně všech relevantních výrobních, skladovacích a přepravních činností je uznávána jako klíčová součást digitální továrny v průmyslu a dnes je široce používána a přijímána. Odborné zdroje uvádí, že v reálných projektech lze dosáhnout snížení zásob a doby průchodnosti o 20 – 60 % a zvýšení produktivity stávajících výrobních zařízení o 15–20 %. To jsou hodnoty, kvůli kterým v dnešní době, kdy je nutno hledat rezervy ve všech oblastech s výrobou a logistikou souvisejících, již stojí zato se zamyslet nad tím, zda počítačovou simulaci výrobních a logistických systémů používat, či nikoli. [1]

1.1 Obecné principy simulace

Simulace v běžném životě může souviset s různými činnostmi. Pokud se toto slovo používá ve výpočetní technice, pak termín simulace představuje proces vytváření abstraktních modelů systému z reálného prostředí a provádění příslušného počtu experimentů na nich. Když jsou experimenty prováděny na počítači, pak se nazývají počítačové modelování a počítačová simulace. [5]

Simulace ve smyslu napodobení systému, jeho stavů a fungování za účelem jeho analýzy spočívá v napodobení reálného systému, nebo procesu a v jeho náhradě vhodným modelem. Tento model může mít různé formy, reálné i virtuální.

Počítačová simulace jakožto forma virtuální využívá výpočetní techniky a vhodných matematických vztahů a algoritmů k dosažení napodobení chování reálného systému, ve vztahu k této práci logistického, nebo výrobního procesu, ale obecně libovolného systému.

V současné době úroveň a výkon výpočetní techniky umožňuje graficky bohaté, dynamické 3D výstupy podobné videohram a s využitím technologie virtuální reality se uživatel může i aktivně účastnit procesu, procházet scénou jako ve videohře a ovlivňovat dění v modelu procesu.

Pro získání nejvýhodnějších ekonomických a technických parametrů pro optimalizaci procesu a maximalizaci výtěžnosti použitých zdrojů ovšem tato bohatá grafická prezentace není tolik potřeba a jistě by dostačovaly i jednodušší např. 2D simulace. Analýza pomocí 3D virtuální reality ale už bude hrát roli např. v případě, kdy bude kladen důraz i na ergonomii procesu s cílem minimalizovat negativní vlivy na zdraví např. pracovníků výrobní linky.

Podstatná je však možnost navrhovat proces rychle a rychle také měnit vstupní parametry např. výrobní linky a získávat rychle výstupy a umožnit urychlení procesu rozhodování, jak požadovaný proces sestavit s nejnižšími nutnými náklady a přitom s nejlepšími možnými ekonomickými výsledky.

Obecně se pro počítačovou simulaci doporučuje následující postup:

- formulace problémů,
- test způsobilosti simulace,
- formulace cílů,
- sběr a analýza dat,
- modelování,
- provedení simulace,
- analýza výsledků a jejich interpretace,
- dokumentace.

1.1.1 Formulace problémů

Formulace problémů je základním požadavkem pro možnost navrhnout počítačovou simulaci. Spolu se zákazníkem musí požadavky na simulaci formulovat odborník znalý obecných principů simulace. Výsledek formulovaného problému by měla být písemná dohoda (např. technická specifikace), která obsahuje konkrétní problémy, které budou studovány pomocí simulace.

1.1.2 Test způsobilosti simulace

Je-li potřeba posoudit způsobilost simulace k řešení určitého problému, je třeba uvažovat více faktorů, které výsledný model ovlivňují.

Mezi tyto faktory patří např.:

- nedostatky analytických matematických modelů (mnoho proměnných),
- vysoká složitost simulovaného systému,
- mnoho faktorů, které je třeba vzít v úvahu,
- nepřesná vstupní data,
- postupné zjišťování limitů systému
- možnost opakovaného použití simulačního modelu.

1.1.3 Formulace cílů

Každá firma se zaměřuje na systém cílů. Ten se obvykle skládá z nějakého zásadního cíle (např. ziskovost), který se pak rozděluje na různé dílčí cíle, které se vzájemně ovlivňují. Definice cílového systému je důležitým přípravným krokem. Časté cíle pro simulaci jsou například:

- zkracování doby zpracování,
- maximalizace využití výrobních prostředků,
- minimalizace zásob,
- urychlení termínů dodání.

Všechny definované cíle musí být sumarizovány a statisticky analyzovány před simulací, což znamená určitou požadovanou úroveň detailů pro simulaci. V důsledku toho pak určují rozsah simulační studie.

1.1.4 Sběr dat

Před simulací je potřeba shromáždit některá potřebná data, např. by mělo jít o tato data:

- data zatížení systému
- organizační údaje
- technická data

1.1.5 Modelování

Fáze modelování zahrnuje sestavení a testování simulačního modelu. Modelování se obvykle skládá ze dvou fází:

- odvození ikonického modelu z konceptuálního modelu.
- přenesení modelu do softwarového modelu.

V první fázi modelování je nejprve třeba získat obecné znalosti o simulovaném systému. Na základě cílů, které mají být testovány, je třeba rozhodnout o přesnosti simulace a na základě přesnosti simulace jsou poté nutná rozhodnutí o tom, které aspekty je třeba zjednodušit. První fáze modelování zahrnuje 2 činnosti:

- analýzu (rozdělení),
- abstrakci (zobecnění).

Pomocí systémové analýzy se složitost systému smysluplně rozloží do jednotlivých prvků v souladu s původními cíli.

Abstrakce, poté sníží množství specifických atributů systému, pokud je to praktické pro vytvoření zjednodušeného modelu. Typickými metodami abstrakce jsou redukce (eliminace nepodstatných detailů) a zobecnění (zjednodušení podstatných detailů).

V druhé fázi modelování je vytvořen a otestován simulační model. Je třeba zahrnout výsledek modelování do dokumentace modelu a provést případné změny simulačního modelu tak, aby byly v souladu s dokumentací. V praxi je tento krok často opomíjen, takže model z důvodu nedostatku dokumentace jeho funkčnosti nelze již znova použít. Proto existuje potřeba komentování modelů a zdrojového kódu během programování, čímž je zajištěno, že popsaná funkčnost je stále srozumitelně popsána a k dispozici i po dokončení modelu.

1.1.6 Provedení simulace

V závislosti na cílech simulační studie se provedou simulace a je realizován zkušební plán. V plánu testování vydávají jednotlivé simulace data, která obsahují proměnné modelu, cíle a očekávané výsledky. Je také důležité definovat časové rozpětí pro simulační experimenty na základě výsledky zkušebních simulací. Simulace může běžet

několik hodin, nebo se kvůli potřebám získání statistických dat simulace mohou i mnohonásobně opakovat. V těchto případech je užitečné zkontrolovat, zda je možné ovládat simulace pomocí samostatných naprogramovaných utilit (dávkové spuštění). Časy realizace simulací lze posunout např. do nočních hodin, takže lze využít dostupnou výpočetní kapacitu optimálně. Vstupní a výstupní data a základní parametry simulačního modelu musí být zdokumentovány pro každý experiment.

1.1.7 Analýza výsledků a jejich interpretace

Správná interpretace výsledků simulace významně ovlivňuje úspěch simulační studie. Pokud jsou výsledky v rozporu s předpoklady je nutné analyzovat, jaké vlivy jsou zodpovědné za neočekávané výsledky. Je také důležité si uvědomit, že složité systémy mají často tzv. náběh fáze. Tato fáze může ve skutečnosti a v simulaci probíhat odlišně. Proto výsledky získané během náběhové fáze jsou často nepřenositelné na modelovaný systém a nemusí mít žádný vliv na hodnocení. V případě, že je potřeba zkoumat i náběhovou fázi reálného systému, musí být tato fáze náběhu systému plně, nebo i samostatně modelována.

1.1.8 Dokumentace

Pro dokumentaci simulační studie se doporučuje forma projektové zprávy. Dokumentace by měla poskytovat přehled o načasování simulace a dokumentovat provedené práce. Zajímavá je v této souvislosti dokumentace neúspěšných systémových variant a konstelací. Jádro projektové zprávy by mělo být prezentací výsledků simulace na základě specifikace požadavku zákazníka. Na základě simulační studie má smysl zahrnout návrhy akcí v dokumentaci. Nakonec lze doporučit popsat simulační model v jeho struktuře funkčnosti.

1.2 Proces počítačového modelování

Informační technologie do značné míry změnily životy lidí a dnes je už téměř nemožné si představit jakoukoli činnost, která je zcela nezávislá na využití výpočetní techniky v nějaké formě, nebo alespoň v některé části celého procesu. Jakmile se objevily první počítačové systémy, lidé se snažili využít počítače k řešení složitých problémů v různých oblastech. S rozvojem průmyslu se vyvíjely požadavky na počítače

a výpočetní software např. ve strukturální analýze, nebo při návrhu výrobků. Kupříkladu tradiční metody konstruování jsou již od konce 80 let, kdy se na trhu začaly objevovat první CAD systémy, nahrazovány počítačovým zpracováním konstrukcí od samotného počátku. Dnešní CAD systémy obsahují softwarové moduly, které jsou schopny simulovat chování konstrukcí při různých podmínkách zatížení. Nákladné experimenty a testy prototypů jsou tedy nahrazovány levnějšími a výkonnějšími výpočetními metodami, které nevyžadují poškození, nebo i zničení celé konstrukce za účelem stanovení její únosnosti, životnosti, ergonomických parametrů a dalších vlastností potřebných ke splnění požadované funkčnosti.

Dle výše uvedeného by tedy nejdříve měla být provedena simulace, obecně jakéhokoli reálného systému a poté, pokud simulace poskytne uspokojivé výsledky, může být provedena realizace návrhu tohoto systému. Počítačová simulace, nebo počítačový model má za úkol simulovat abstraktní model určitého nebo ekvivalentního systému. Počítačové simulace se staly užitečnou součástí matematického modelování ve všech oborech inženýrství, stejně jako modelování mnoha přírodních systémů ve fyzice, mechanice, chemii, biologii, ekonomických systémech, psychologii a společenských vědách, aby bylo možné získat lepší přehled o funkčnosti výše uvedených systémů.

Aby byl model užitečný, je nutné určit jeho chování pro definovanou a omezenou množinu proměnných. To znamená, že pro některé náhodné vstupní parametry jsou pozorovány odpovídající výstupní hodnoty.

Pokud vezmeme v úvahu modelování a simulace, vstupní data se budou lišit a záviset na mnoha faktorech. Některé modely si vystačí s velmi jednoduchými vstupy (např. skalární veličiny, teplota, tlak, jednorozměrné elektrické veličiny apod.), zatímco jiné modely naopak vyžadují až terabajty vstupních dat (např. simulace počasí, klimatických změn, modely chování lidské společnosti). [5]

Vstupní data pro simulaci mohou být získávána z různých čidel, vstupních zařízení, nebo dříve získaných uložených dat, těmito zdroji mohou být:

- senzory a další fyzická zařízení, která jsou připojena k modelu,
- ovládací panel, který nějakým způsobem přímo ovlivňuje průběh samotné simulace,
- aktuální nebo starší data přenesená ručně,
- hodnoty, které představují výstupní produkty z jiných procesů nebo operací,
- hodnoty, které představují výstupní prvky jiných modelů nebo simulací.

1.3 Vhodné případy pro využití počítačové simulace

Zásadní otázkou by mohlo být, jestli je opravdu nutno používat počítačové modelování a počítačovou simulaci. Tato otázka je kladena většinou v okamžiku, kdy počítačová simulace ještě není zavedenou součástí organizace, nebo podniku a její plánované zavedení zcela určitě přinese nemalé náklady na veškeré počítačové vybavení, lidské zdroje a všechny další náležitosti s tím související. Existuje více důvodů kdy a proč dát přednost počítačové simulaci před dřívějšími konvenčními postupy. [5] Nejdůležitější důvody by mohly vznikat např. v následujících situacích:

- nelze určit analytické řešení analytického modelu,
- prototyp, konstrukce, nebo obecně systém je příliš složitý a nelze jej analyticky popsat,
- experiment v reálném systému nebo experiment na reálném systému je buď nerentabilní, nebo příliš složitý,
- pokud je nutné simulovat podmínky, které by vedly ke zničení systému a zničení skutečného systému není možné, je počítačová simulace v takových situacích jediným řešením.

Je mnoho dalších důvodů, kvůli kterým je vhodné použít počítačovou simulaci. Cílem modelování a simulace je často vnímat funkčnost stávajícího reálného systému, jehož struktura je sotva známá nebo se k ní nelze přiblížit. Když je potřeba optimální nebo optimalizované fungování systému, je nutné změnit různé parametry. Pokud se vezme v úvahu skutečný systém, je to často nemožné, protože takový systém neexistuje. Jinými slovy, takový systém ještě nebyl postaven, nebo ceny takového experimentu jsou extrémně vysoké. V takových situacích jsou nejlepším řešením modelování a simulace.

Pokud jde o dlouhodobé procesy reálného systému nebo uvnitř reálného systému, pak může být problematickým faktorem čas. V takových situacích může počítačová simulace proces urychlit a potřebný čas zkrátit. U extrémně rychlých procesů reálného systému nebo v rámci reálného systému je řešením počítačová simulace, která umožňuje sledovat vysokorychlostní procesy postupně nebo pomalu. To je velmi důležité, protože to není možné v reálném životě nebo v reálném prostředí. Někdy by měl být experiment z různých důvodů zastaven a často je to v reálných podmínkách

nemožné. Pokud jde o počítačovou simulaci takového experimentu, není problém, protože simulaci lze zastavit a v případě potřeby pokračovat. [5]

1.4 Výhody a nevýhody počítačové simulace

Ani počítačové simulace nejsou dokonalé a existují různé problémy. Ač jsou obecně velmi užitečné, mají své výhody i nevýhody. Základní výhody počítačových simulací jsou:

- když je model vytvořen, lze jej opakovaně použít pro analýzu požadovaných procesů, struktur a podobných prvků,
- počítačové simulace mohou být velmi užitečné, i když jsou vstupní data neúplná a vykazují určitou míru nepřesnosti,
- ve většině případů je jednodušší a levnější získat výstupní data simulace než výstup dat skutečného systému,
- v některých případech může být počítačová simulace jediným způsobem, jak správně vyřešit problémy.

Mezi hlavní nevýhody počítačových simulací patří:

- tvorba simulačních modelů i počítačových simulací může být nákladná a časově náročná (jedná se o čas potřebný pro jejich vývoj, testování a ověřování),
- pomocí počítačových simulací nelze získat vztah mezi výstupními a vstupními proměnnými ani optimální řešení,
- pro vývoj a použití simulačních modelů a počítačových simulací je nutná znalost různých nástrojů a metod,
- vyhodnocení modelu je poměrně složitý proces a vyžaduje další experimentování v různých prostředích.

Nehledě na výše uvedené nevýhody, je počítačová simulace velmi užitečná technika zpracování návrhu, nebo analýzy systému a její využití rychle narůstá v podmínkách a situacích, kde je to možné. Ve skutečnosti použití počítačové simulace má mnohem více výhod než nevýhod a je jisté, že počítačová simulace bude v budoucnu dominantní téměř v každé oblasti a prostředí každodenního života. [5]

2 Analýza databází vhodných k použití s Plant Simulation

Z hlediska způsobu ukládání dat a vazeb mezi nimi, je pro účel této práce pod pojmem databáze uvažována relační databáze a použití jazyka SQL pro dotazy na databázi, manipulaci s daty a další potřebné operace.

Pojmem databáze se často označují různé nástroje, softwarové komponenty, jednotlivé logické vrstvy databázového systému i databázový systém (DBS) jako celek. Jednotlivé vrstvy mohou být realizovány jako odděleně spustitelný software, nebo mohou být sloučené do jednoho celku.

Pro účely této práce je potřeba tyto termíny rozlišovat a obecný pojem databáze je zde asi nejlépe použít pro databázový systém jako celek. Pro použití v dalším textu práce jsou následovně vymezeny některé pojmy:

- **datové úložiště** – databázový stroj (úložný stroj, Database Engine, Storage Engine), základní databázová komponenta (vrstva) pro vytváření, čtení, editaci a mazání (CRUD) dat z databáze a pro ukládání dat, zde chápáno jako softwarová komponenta pojmově sloučená s fyzickou vrstvou pro samotné ukládání dat na paměťové médium,
- **DBMS** – systém řízení báze dat (SŘBD), software, zajišťující práci s datovým úložištěm, rozhraní mezi uživatelskými aplikacemi a uloženými daty,
- **ODBC, JDBC** – Open Database Connectivity, Java Database Connectivity softwarové API, rozhraní mezi DBMS a uživatelskou aplikací pro přístup k DBMS nezávislý na operačním systému, programovacím jazyku a databázovém systému,
- **uživatelské administrační rozhraní DBMS** – uživatelské rozhraní (textové, nebo grafické) pro ovládání DBMS. Může být realizováno jako samostatná komponenta oddělená od DBMS, nebo může být součástí DBMS. Ale jako minimální možnost administrace DBMS bývá součástí DBMS alespoň textové konzolové rozhraní,
- **databáze, databázový systém** - celek zahrnující DBMS zároveň s bází dat, tzn. včetně datového úložiště a včetně uživatelského administračního rozhraní DBMS, ať už jsou tato realizována jako jediný celek, nebo samostatné komponenty, nebo např. dvě některé vrstvy sloučené a další samostatná.

Všechny výše uvedené pojmy chápané jako virtuální logické, nebo fyzické vrstvy by bylo možno v některých případech dále dělit, nebo slučovat, neboť různé databázové systémy se liší podle své konkrétní implementace jednotlivých vrstev. Např. datové úložiště jako komponenta by šlo jako vrstva dále dělit podle své softwarové části a podle konkrétního systému ukládání dat, nebo indexů do souborů na paměťové médium, neboť např. data a indexy mohou být ukládány zvlášť do samostatných souborů, ale mohly by být ukládány i do společného souboru, nebo může být implementace fyzické vrstvy navržena nějak jinak.

Také je možno pro různá datová úložiště používat různé DBMS a naopak různé DBMS mohou používat různé databázové stroje. Např. MySQL a MariaDB mohou používat databázové stroje InnoDB a MyISAM a MySQL mimo tyto ještě cca 6 dalších.

Samostatným druhem software jsou pak DBC konektory pro připojení k databázím, které slouží pro připojení a správu databází pomocí software třetí strany, nebo vlastním uživatelským software.

Z pohledu výše uvedeného rozdělení se v obecné terminologii používají další výrazy, které se v této práci vyskytnou a mohou mít více významů. Jsou to např.:

- **databázový server** - lokální, nebo vzdálený (síťový) počítač, který poskytuje datové služby, nebo program, který tyto služby realizuje. Jako databázový server tedy může být označován nejen hardware ale i software. V případě software může tento běžet na stroji jako jeden z mnoha dalšího instalovaného software např. v uživatelském PC, nebo může být pro server vyhrazen samostatný stroj,
- **webový server** – jako v případě DB serveru se taktéž může jednat o hardware i software a platí zde i ostatní skutečnosti jako pro DB server.

Z tohoto pohledu se jedná o obecné termíny, které je třeba specifikovat v konkrétním významu.

Většina systémů pro správu databází obsahuje vlastní API, které umožňuje uživateli komunikovat s databázovým úložištěm, aniž by použil uživatelské rozhraní DBMS. Aplikace tedy mohou k datům přistupovat a manipulovat s nimi přes API a uživatelské rozhraní DBMS může častěji sloužit pro údržbu dat.

2.1 Tecnomatix Plant Simulation

Tecnomatix Plant Simulation je simulační software a součást balíku digitálních řešení určených převážně pro výrobu a průmyslové podniky od společnosti Siemens Product Lifecycle Management Software Inc. Přes toto primární zaměření lze ale SW tohoto druhu použít pro téměř všechny obory lidské činnosti, neboť i státní instituce, vysoké školy, nemocnice a další velké organizace by mohly pomoci takového SW navrhovat, simulovat a ladit své např. logistické procesy.

Na webu společnosti Siemens PLM Software se o tomto SW říká: „*Tecnomatix je komplexní portfolio digitálních výrobních řešení, která vám pomohou digitalizovat výrobu a proces přeměny vašich inovativních nápadů na transformativní produkty. Se softwarem Tecnomatix dosáhnete synchronizace mezi produktovým inženýrstvím, výrobním inženýrstvím, výrobou a servisními operacemi, abyste maximalizovali svou efektivitu.*“ [1]

Tecnomatix Plant Simulation je součástí širšího navazujícího balíku nástrojů z kterých může brát vstupy a do kterých může předávat výstupy a umožňuje velmi pokročilou vizualizaci včetně 3D simulace a i virtuální průchod modelovaným prostředím, např. výrobní halou, nebo skladem (Obr. 2.1).



Obr. 2.1 Ukázka simulace procesu v Tecnomatix Plant Simulation [4]

V produktech Siemens Tecnomatix lze také uživatelsky vytvářet softwarová rozšíření. Pro aplikace Process Simulate, Process Designer, RobotExpert, nebo ShopFloor lze využít .NET API. Jako většina podobných nástrojů, i Tecnomatix Plant Simulation

modeluje objekty a entity podobné reálnému světu, a každý objekt v modelu má možnost mít vlastnosti a stavy tomu odpovídající.

Tecnomatix Plant Simulation používá pro rozšíření možností modelování a ovládání simulace procesů interní jazyk SimTalk. Každý objekt v modelu má vestavěné vlastnosti poskytující mnoho užitečných funkcí. Programátorsky zkušenější uživatel může použít i jazyk C, ale použití SimTalk je uživatelsky snazší, mj. i proto, že k programování není potřeba samostatné IDE, neboť toto je součástí Plant Simulation.

2.2 DBMS – nástroje na správu relačních databází

Jelikož správa databází je možná více způsoby a nástroji jak výrobců databází, tak i dodavatelů třetích stran a v různých kombinacích, je lépe přiblížit si tyto nástroje odděleně. Některé nástroje pro správu relačních databází umožňují připojení a správu více databází od různých výrobců, některé mohou být jednocelové. Pro účel této práce je zmíněn jen výběr několika nástrojů v práci použitých, ale na trhu je dostupných jistě mnoho dalších ať už zdarma, nebo zpoplatněných, open source i proprietárních.

2.1.1 Microsoft Office Access

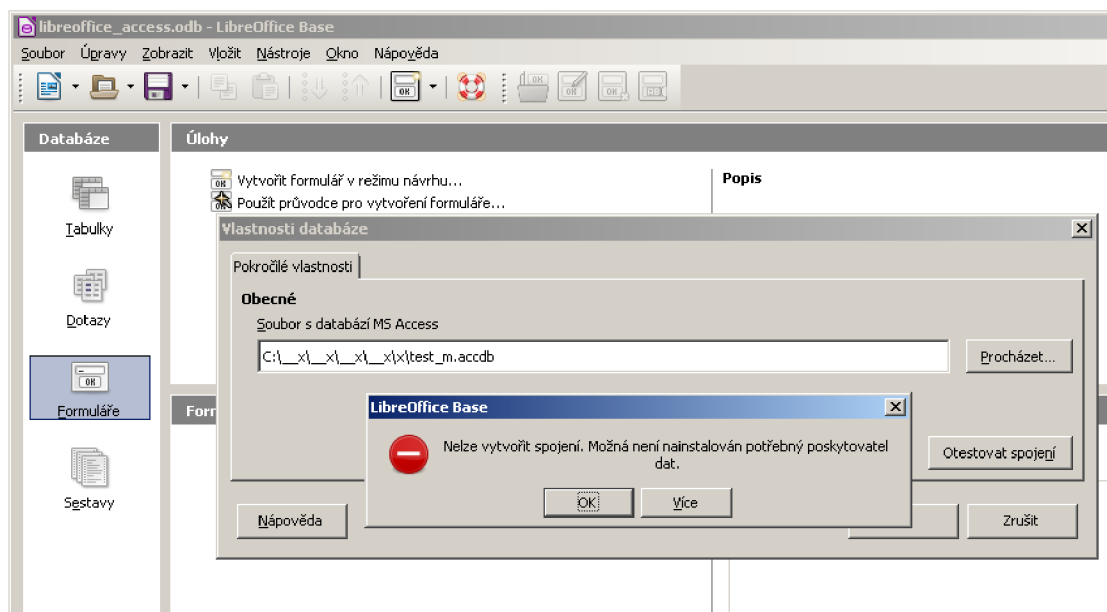
Microsoft Office Access je nástroj na správu relačních databází od společnosti Microsoft, který je typicky součástí Microsoft Office a kombinuje relační Microsoft Jet Database Engine s grafickým uživatelským rozhraním. Umí přistupovat k datům z Access/Jet, Microsoft SQL Server, Oracle či ke kterékoliv další databázi přes rozhraní ODBC. [8]

MSO Access jako součást balíku MS Office nabízí uživatelům mnoho zajímavých možností, jak i pracovníci, kteří nejsou na IT pozicích mohou přistupovat k firemním a obecně k různým datům nástrojem, který mají v PC většinou už předinstalován. MSO Access nabízí poměrně snadné řešení tam, kde by mohlo být zbytečné nasazovat dražší komplexnější, rozsáhlejší a i na správu a údržbu složitější serverové produkty, jako jsou MS SQL server, nebo Oracle Database, které by se např. menší firmě nevyplatily, nebo by byly vzhledem k jednoduchým požadavkům na databázi zbytečně komplikované.

2.1.2 LibreOffice / Apache OpenOffice

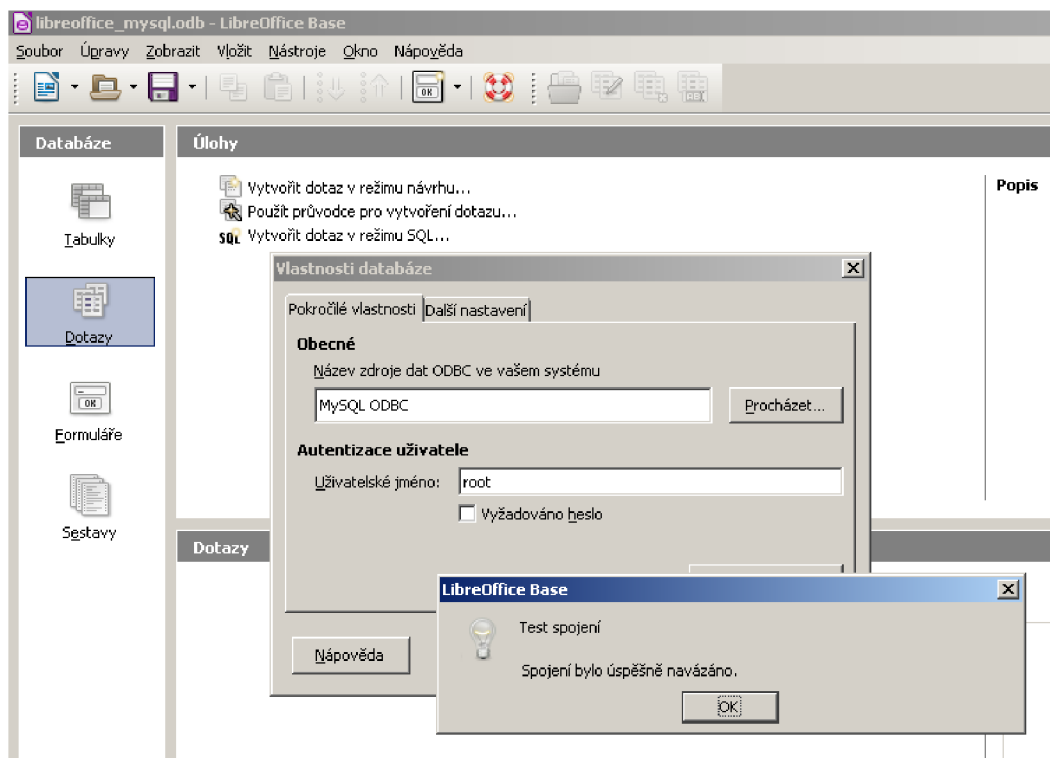
Kancelářské balíky LibreOffice a Apache OpenOffice (dříve OpenOffice.org) jsou následníky kancelářského balíku StarOffice, který v r. 1999 zakoupila společnost Sun Microsystems (od roku 2010 součástí firmy Oracle) s cílem vytvořit srovnatelnou alternativu k Microsoft Office. V r. 2010 skupina předních vývojářů z projektu odešla a založila The Document Foundation, která převzala vývoj tohoto kancelářského balíku pod názvem LibreOffice. V r. 2011 firma Oracle ukončila vývoj OpenOffice.org a předala ho Apache Software Foundation, kde je nadále vyvíjen jako Apache OpenOffice. [9]

Tyto kancelářské balíky jsou dnes asi nejrozšířenější bezplatnou variantou k balíku Microsoft Office a podobně jako MSO, nabízejí také databázovou aplikaci podobnou MSO Access. Ovšem ze staršího Apache OpenOffice se nepodařilo připojit ani k MariaDB, ani k MSO Access (Obr. 2.2).

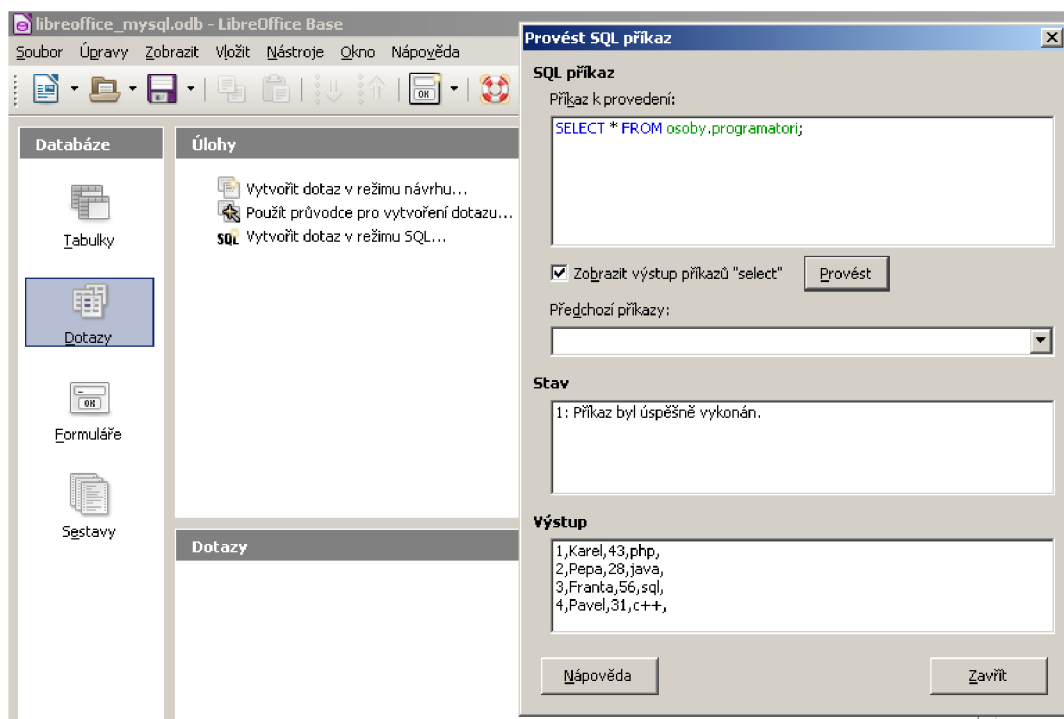


Obr. 2.2 Pokus o připojení k souboru MSO Access z LibreOffice přes ODBC driver pro MSO Access – vlastní zpracování

Z novějšího LibreOffice se sice podařilo připojení k MariaDB (Obr. 2.3), ale funkčnost se nezdá úplná. Bylo možno vykonat SQL příkaz SELECT (Obr. 2.4), ostatní příkazy nebyly úspěšné. K souboru MSO Access (formát ACCDB) se nepodařilo připojit vůbec. Je možné, že spolu nekomunikují ODBC ovladače různých verzí a různě staré verze Apache OpenOffice / LibreOffice a soubor ACCDB z MSO verze 2010.



Obr. 2.3 Test připojení k databázi MariaDB z LibreOffice přes ODBC driver pro MySQL – vlastní zpracování

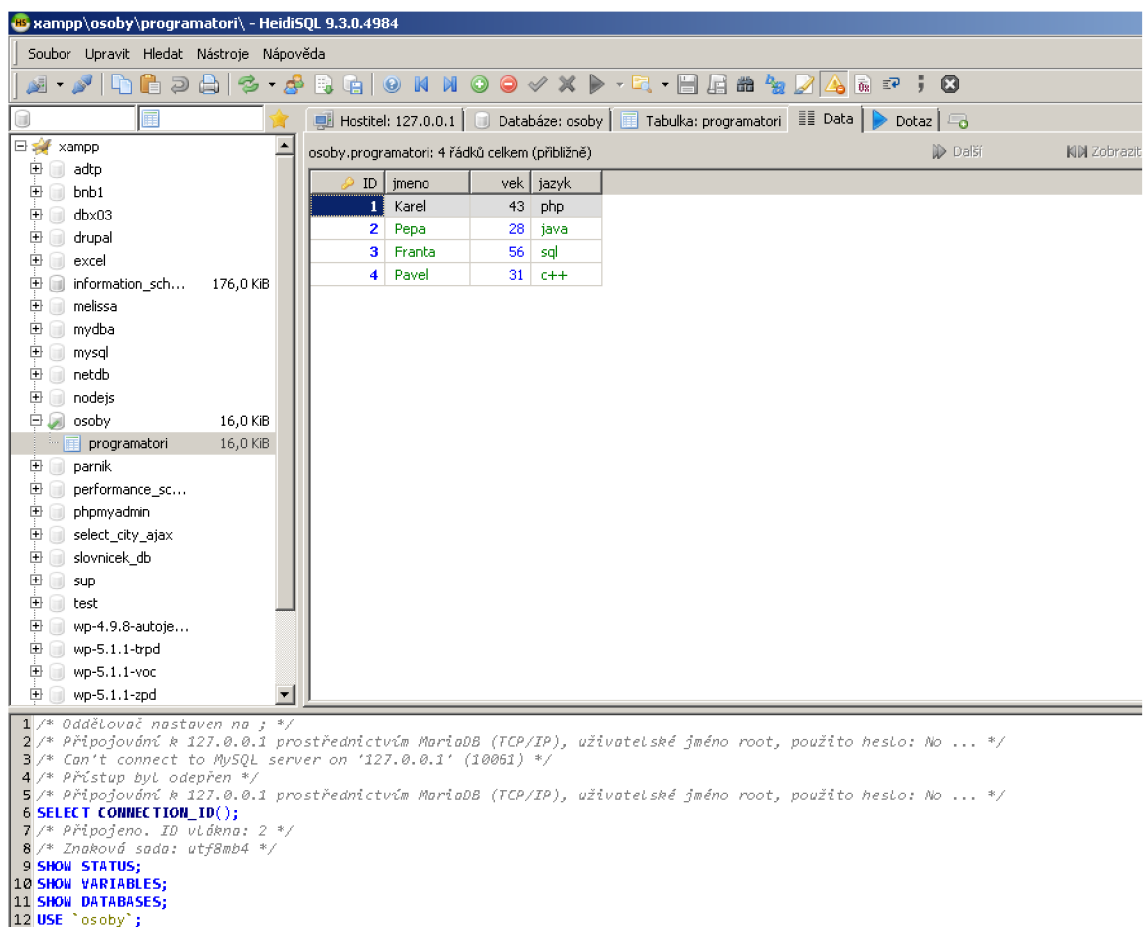


Obr. 2.4 Pokusný dotaz na databázi MariaDB z LibreOffice - vlastní zpracování

2.1.3 HeidiSQL

HeidiSQL je freeware umožňující spravovat databáze MariaDB, MySQL, Microsoft SQL, PostgreSQL a SQLite. Autorem je Ansgar Becker a vývoj HeidiSQL probíhá od roku 2002. [10]

Z hlediska uživatele je největší přínos HeidiSQL v tom, že může být vhodnou, jednoduchou a uživatelsky přívětivou variantou pro správu databází v lokálním umístění a výhodou je i česká jazyková lokalizace (Obr. 2.5).



Obr. 2.5 Prostředí aplikace HeidiSQL - vlastní zpracování

2.1.4 DB Browser for SQLite (DB4S)

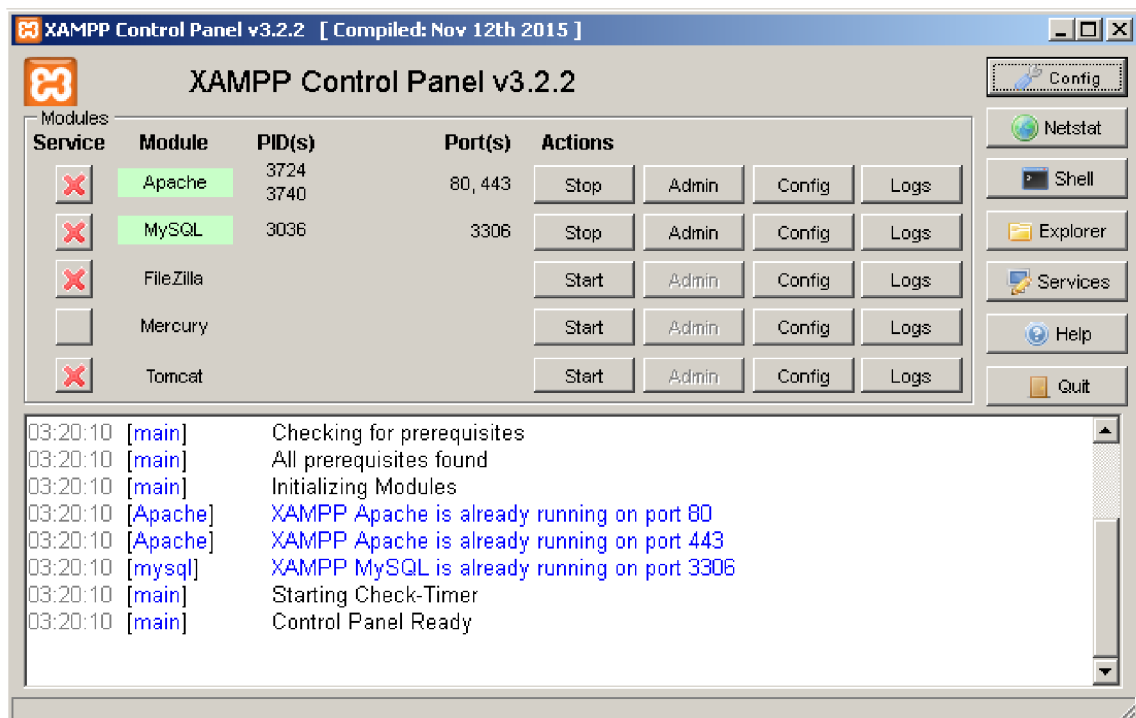
DB Browser for SQLite je vizuální nástroj s otevřeným zdrojovým kódem pro vytváření, navrhování a úpravy databázových souborů kompatibilních s SQLite. DB4S používá známé rozhraní podobné tabulkovému procesoru. [11]

DB4S je jednoduchý a uživatelsky přívětivý nástroj správu pro SQLite databází, výhodou je také i česká jazyková lokalizace.

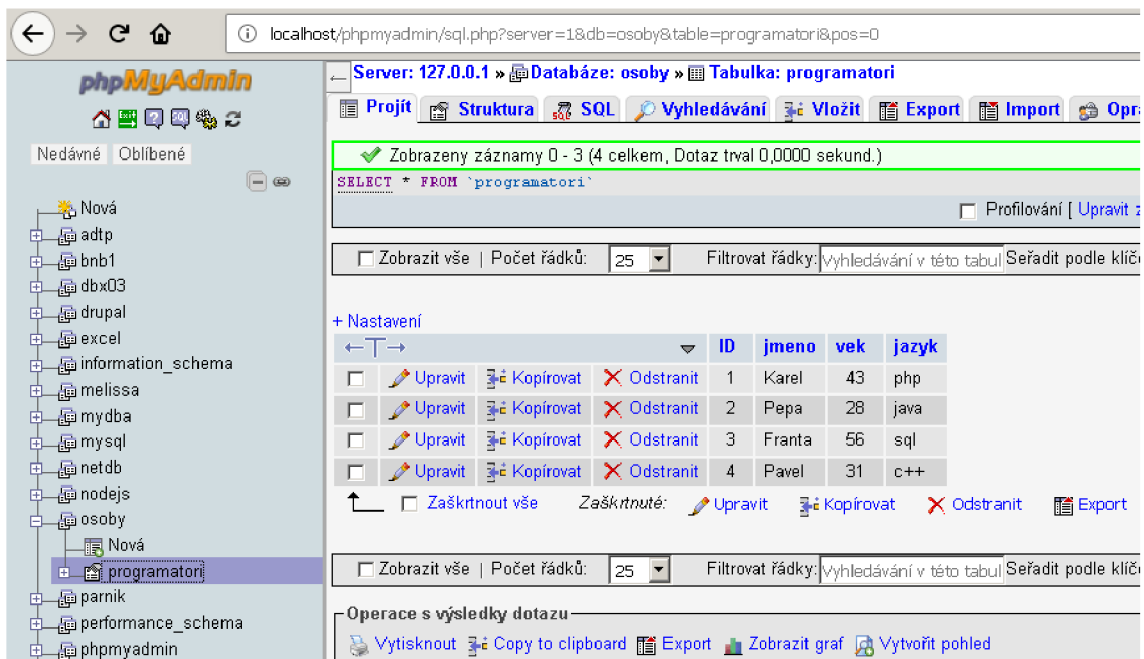
2.1.5 phpMyAdmin

phpMyAdmin je bezplatný nástroj napsaný v jazyce PHP umožňující jednoduchou správu obsahu databází MariaDB a MySQL prostřednictvím webového rozhraní. Umožňuje provádět širokou škálu operací, jako je správa databází, tabulek, sloupců, vztahů, indexů, uživatelů, oprávnění, provádět SQL příkazy atd. Je k dispozici ve více jak 70 jazycích. [12]

Správa webových databází externím nástrojem umístěným lokálně v PC uživatele většinou vyžaduje zvláštní svolení a nastavení od provozovatele webhostingu a proto je nejčastěji používaným nástrojem pro správu webových databází MariaDB a MySQL právě phpMyAdmin, který je v dnešní době provozovateli webhostingu nabízen již předinstalován v ceně webhostingu jako standard a jako hlavní nástroj pro správu databází umístěných na webhostingu. Pro lokální využití databází MySQL a MariaDB je vhodným řešením instalace balíku, například XAMPP, jehož obsahem je webový server (Apache), databáze (MariaDB) a další aplikace (Obr. 2.6). MySQL a MariaDB lze instalovat a spouštět i samostatně, ale instalací balíku XAMPP uživatel získá nejen databázi, ale i nástroj pro správu databází phpMyAdmin (Obr. 2.7). Jelikož se ovšem jedná o webovou aplikaci, potřebuje phpMyAdmin pro svůj běh mít spuštěn webový server, v tomto případě server Apache.



Obr. 2.6 Prostředí aplikace XAMPP - vlastní zpracování

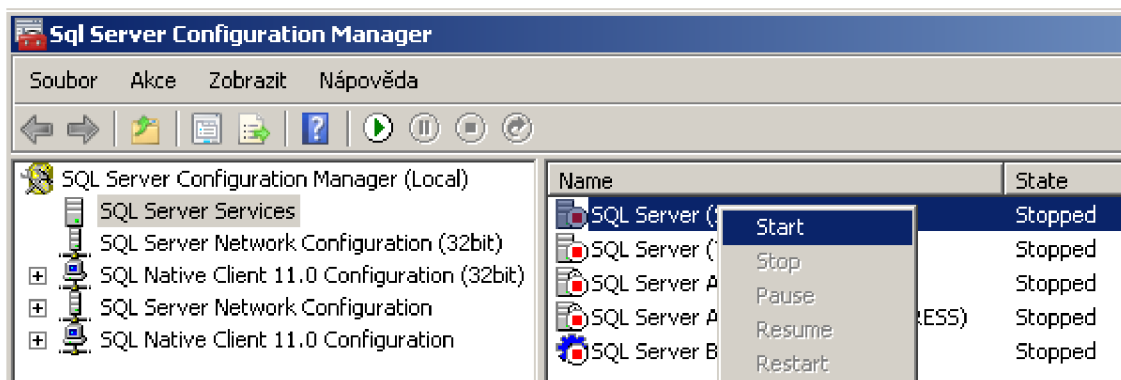


Obr. 2.7 Prostředí aplikace phpMyAdmin a připojení k databázi MariaDB - vlastní zpracování

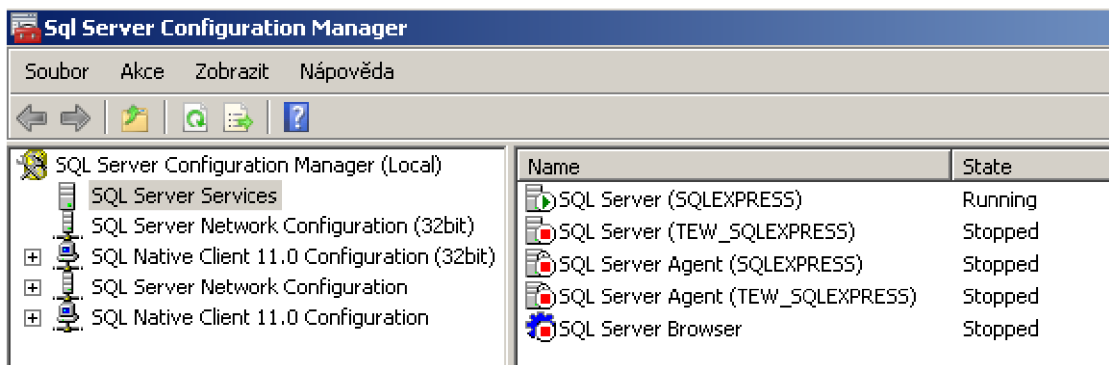
2.1.6 Microsoft SQL Server Management Studio

Pro potřeby této práce byl použit MS SQL Server Express, což je varianta Microsoft SQL Serveru, která je k volně dispozici pro stažení, užití i redistribuci. Je určena pro menší aplikace, které potřebují pracovat s daty uloženými v SQL databázi. Vychází z Microsoft SQL Server Data Engine (MSDE). [13]

S instalací MS SQL Server Express se nainstaluje více nástrojů pro práci s SQL serverem. Samotný databázový stroj se spouští jako služba OS MS Windows, ať už v nástroji služby operačního systému, nebo nástrojem SQL Server Configuration Manager (Obr. 2.8 a 2.9), který je součástí instalace SQL Serveru.

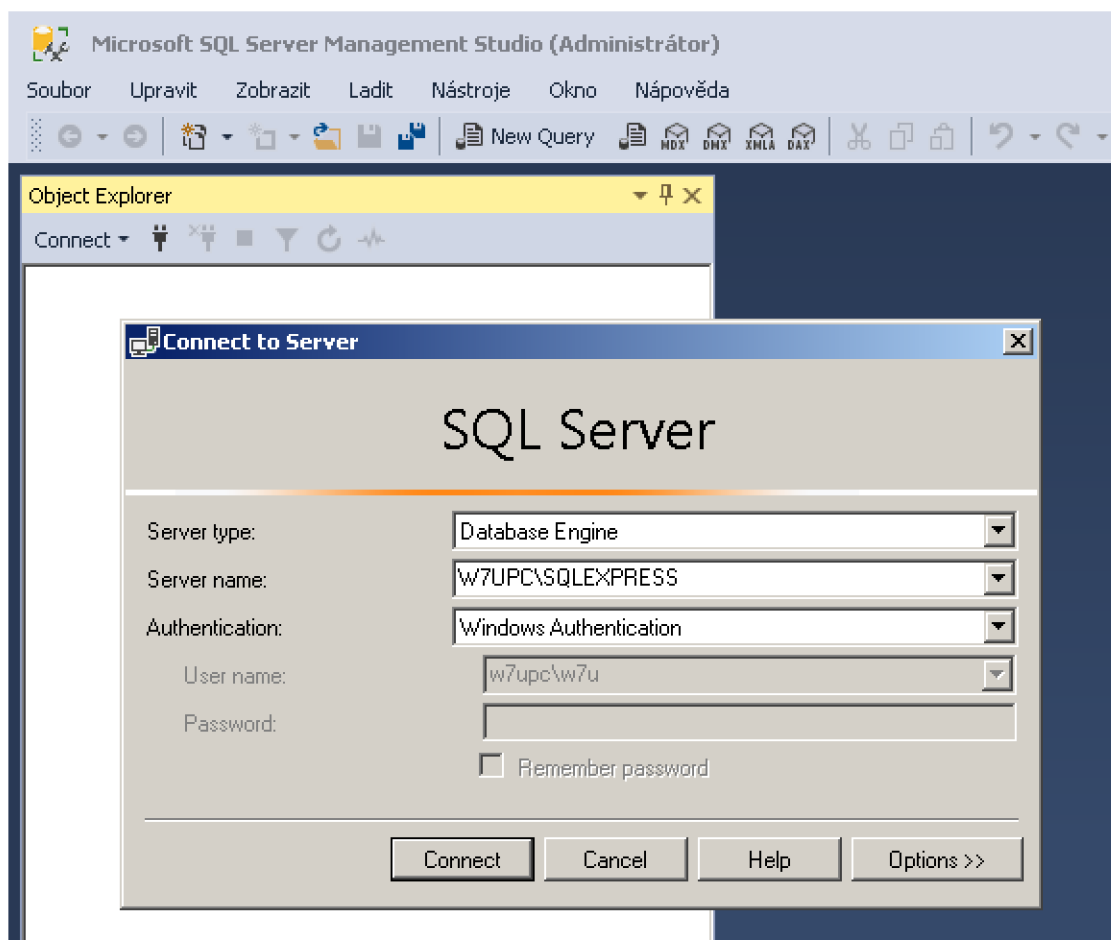


Obr. 2.8 SQL Server Configuration Manager, spuštění SQL Serveru - vlastní zpracování

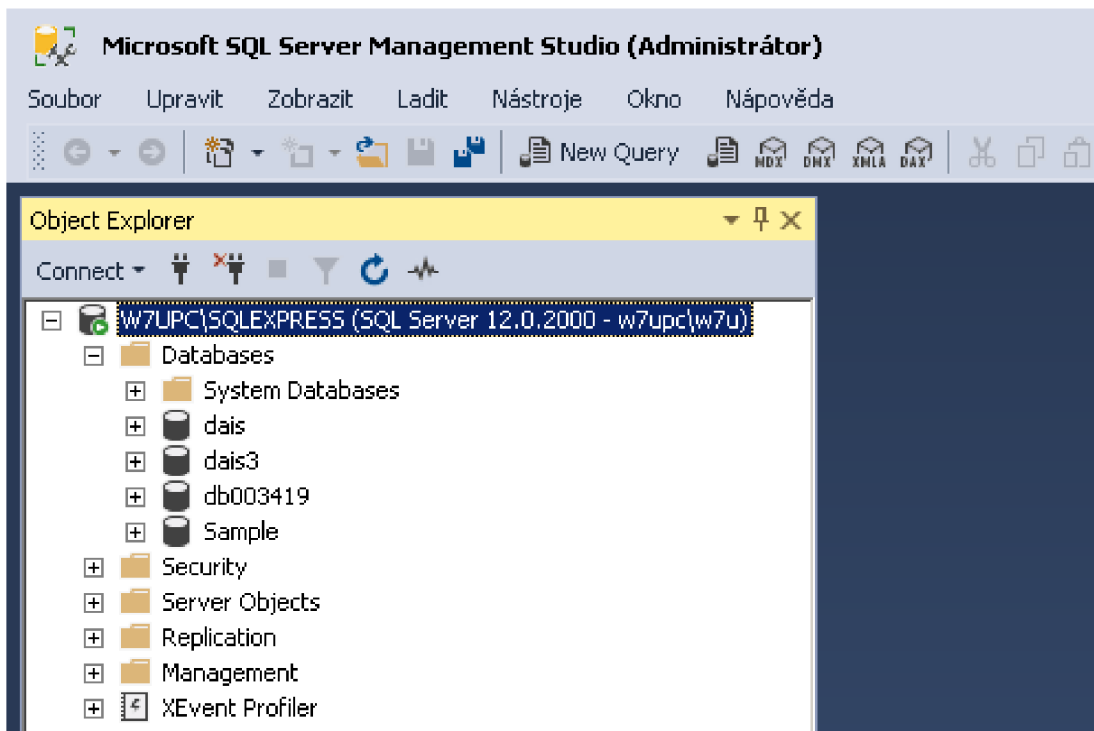


Obr. 2.9 SQL Server Configuration Manager, spuštění SQL Serveru - vlastní zpracování

Po spuštění databázového stroje je možno se k němu připojit nástrojem SQL Server Management Studio (Obr. 2.10 a 2.11), taktéž dodávaným a instalovaným zároveň s balíkem MS SQL Serveru. Tento pak umožňuje správu databází běžících na MS SQL Serveru.



Obr. 2.10 Microsoft SQL Server Management Studio, připojení k SQL Serveru - vlastní zpracování



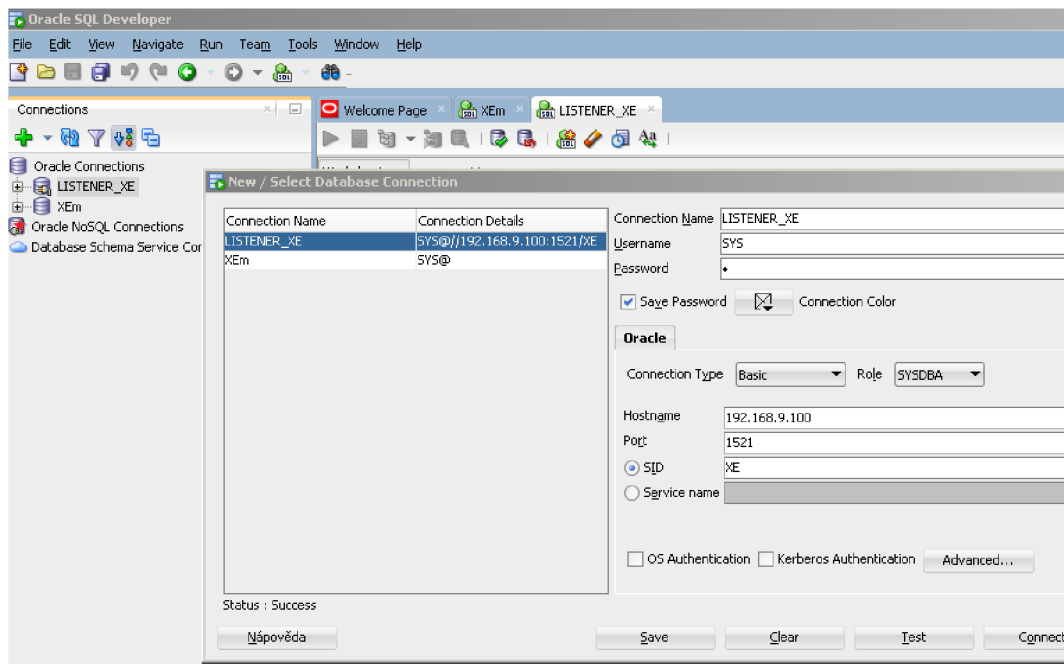
Obr. 2.11 Microsoft SQL Server Management Studio, připojení k SQL Serveru - vlastní zpracování

2.1.7 Oracle SQL Developer

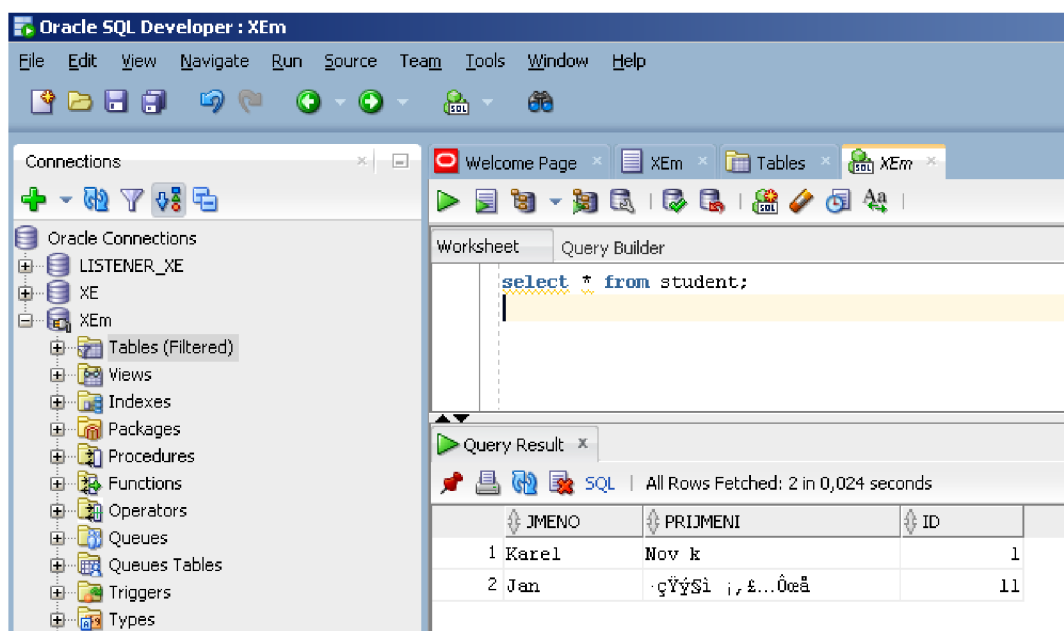
Oracle SQL Developer (Obr. 2.12 a 2.13) je bezplatné IDE a grafické uživatelské rozhraní pro práci s SQL v databázích Oracle. Oracle Corporation poskytuje tento produkt zdarma.

Oracle SQL Developer podporuje produkty Oracle. V minulosti byla podporována řada pluginů třetích stran, které mohli uživatelé nasadit pro připojení k databázím jiných než Oracle. Oracle SQL Developer umožňuje pracovat s databázemi IBM Db2, Microsoft Access, Microsoft SQL Server, MySQL, Sybase Adaptive Server, Amazon Redshift a Teradata. [3] a běží na jakémkoli operačním systému, který podporuje Javu.

SQL Developer poskytuje výkonné editory pro práci s SQL, PL/SQL, uloženými procedurami Java a XML. Umožňuje spouštění dotazů, generování exportů dat do požadovaných formátů (XML, Excel, HTML, PDF atd.) a mnoho dalšího. [15]



Obr. 2.12 Prostředí aplikace Oracle SQL Developer a připojení k databázi - vlastní zpracování



Obr. 2.13 Oracle SQL Developer dotaz na databázi Oracle - vlastní zpracování

2.3 Relační databáze pro použití s Plant Simulation

Relačních databází se dnes nabízí celá řada, v níže uvedeném textu jsou popsány ty nejrozšířenější, u nichž je předpoklad, že by mohly pokrýt většinu průmyslových použití.

2.2.1 SQLite - interní implementace v Plant Simulation

První nejpřirozenější možností jak využívat relační databázi v Plant Simulation je interní implementace engine databáze SQLite přímo v Plant Simulation. Tím se také jeví zbytečnou použití SQLite jako externí databáze. Navíc použití SQLite je možné i v nejnižší dostupné licenci (Education). I začátečník, který potřebuje s Plant Simulation teprve získávat zkušenosti, může i v nejnižší dostupné licenci využívat relační databázi.

2.2.2 MSO Access

DBMS MSO Access se jeví jako výhoda pro např. menší firmy a ostatní uživatele, kteří nemají, nebo nechtějí používat velké databázové systémy jako MS SQL Server, nebo Oracle Database. Jelikož je dostupný v kancelářském balíku MS Office, dá se zařadit mezi snadno dostupná řešení. MSO Access ovšem proti větším a lépe vybaveným databázím nenabízí možnosti procedurálních rozšíření jazyka SQL a ani některé obvyklé příkazy. Např. příkaz TRUNCATE pro vyprázdnění tabulky. Ten nemusí být zrovna vždy žádoucí, ale v případě použití MSO Access je potřeba počítat s některými specifickými omezeními a tato omezení případně obejít nějakým náhradním způsobem. Jak je uvedeno v předešlých kapitolách. Existují také samostatné databázové konektory, pomocí kterých je možno např. v uživatelské aplikaci číst databázové soubory formátu MSO Access a zapisovat do nich, aniž by bylo potřeba na uživateli PC mít instalován MSO Access. To se opět může jevit jako zjednodušení při požadavku na přenositelnost, nebo na mobilitu řešení.

Připojení k Plant Simulation je nutno provést přes ODBC.

2.2.3 MySQL / MariaDB

Databáze MySQL a z ní vycházející odnož MariaDB jsou častým řešením pro použití jako webové databáze. MySQL koupila a dále vyvíjí společnost Oracle, MariaDB je pak

pokračováním vývoje od původních autorů MySQL, jehož cílem je udržet databázi i nadále pod licencí GNU GPL. Výhoda použití těchto systémů by se mohla projevit při požadavku na připojení Plant Simulation k webovému informačnímu systému. V dnešní době je mnoho webových systémů, postavených na platformě Linux, Apache / Nginx, PHP, MySQL / MariaDB, nebo by byla možná i jakákoli jiná kombinace OS, server, jazyk aplikační logiky, databáze a MySQL / MariaDB lze použít i v libovolné kombinaci s jinými OS, servery a jazyky.

Druhou častou webovou konfigurací je MS Windows, MS Windows Server, .NET a MS SQL server. I v této kombinaci by se ovšem na „konci řetězce“ dala použít databáze MySQL / MariaDB.

Připojení k Plant Simulation je nutno provést přes ODBC.

2.2.4 Microsoft SQL Server Express

Microsoft SQL Server patří k nejrozšířenějším databázovým systémům pro správu relačních databází. Jedná se o databázový server s primární funkcí ukládání a získávání dat podle požadavků jiných softwarových aplikací, které mohou běžet buď na stejném počítači, nebo na jiném počítači v síti, včetně internetu. Microsoft uvedl na trh přes desítku různých edic MS SQL Server, které jsou zaměřeny na různé cílové skupiny a pro pracovní zatížení od malých aplikací na jednom počítači až po velké internetové aplikace s mnoha souběžnými uživateli. SQL Server Express je varianta relační databáze Microsoft SQL Serveru, která je k volně dispozici pro stažení, užití i redistribuci. Je určena pro menší aplikace, které potřebují pracovat s daty uloženými v SQL databázi [14]

2.2.5 Oracle Database

Oracle Database je systém řízení báze dat vyvinutý společností Oracle Corporation. Jde o multiplatformní databázový systém s pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností. Stejně jako MS SQL Server i Oracle Database patří k nejrozšířenějším databázovým systémům. Jeho výhodou je navíc to, že Plant Simulation obsahuje nativní připojení k Oracle DB formou samostatného modulu a není potřeba dalších ODBC ani jiných nástrojů k připojení databáze. Je akorát potřeba mít některou z vyšších licencí. [15]

2.4 Shrnutí analýzy databází vhodných pro Plant Simulation

Z výše uvedené analýzy databází vhodných pro použití s Plant Simulation neplyne jednoznačný výsledek, co se týče nějaké lepší, nebo horší vhodnosti použití. Všechny uváděné databáze a DBMS jsou přiměřeně vhodné podle požadovaného způsobu využití a bude záležet určitě např. více na tom, kterou z databází již ten který podnik využívá, nebo pro kterou variantu se z libovolného důvodu rozhodne. Pro každou relační databázi je ovšem také potřeba ovládat příslušnou podmnožinu jazyka SQL, respektive pak pro MS SQL Server a Oracle DB jejich proprietární rozšíření jazyka, tedy T-SQL pro MS SQL Server a PL/SQL pro Oracle DB.

Plant Simulation nabízí 1 interní implementaci databáze pro případ, kdy uživatel nemůže, nebo nechce použít externí databázi, pro připojení k DB Oracle nabízí 2 samostatné databázové objekty pro 2 různé verze Oracle DB a pro všechny ostatní databáze nabízí připojení přes ODBC. Tam už pak záleží na uživateli, pro kterou DB si ODBC ovladače nainstaluje a kterou DB použije.

Pokud by šlo o malý projekt, vystačí si uživatel s Plant Simulation interní DB SQLite, přes ODBC se může připojit k MSO Access, pokud by šlo o webový projekt, může použít MySQL / MariaDB, pro velký projekt a velká data, je k dispozici MS SQL Server přes ODBC, nebo Oracle DB přes databázové připojovací objekty, které jsou součástí Plant Simulation.

Uživatel má tedy v případě nasazení Plant Simulation všechny v současné době běžné a obvyklé možnosti a dá se říci, že z hlediska Plant Simulation není ve výběru databáze nijak omezen. Pravděpodobnost, že by byl uživatel nucen použít nějakou velmi neobvyklou databázi a že by byly problémy s připojením k PtS je tedy poměrně malá.

3 Programování v SimTalk pro využití databází v Plant Simulation

Pro využití databází při práci s PtS je potřeba se k databázi nejprve připojit. PtS nabízí v principu 2 základní možnosti, jak databáze používat. První možností je využít implementovanou interní databázi SQLite a druhou možností je připojení k nějaké externí databázi. Použití externí databáze se ještě dále dělí na možnost použití databáze Oracle, kde je nabídka 2 objektů pro 2 verze DB Oracle a 1 objektu pro obecné připojení k ostatním externím databázím přes ODBC. Pro připojení k databázi přes ODBC je pak v principu jedno, jestli se jedná o databázi lokální (např. MS Access), nebo síťovou. Tzn. např. i MS SQL Server je možno mít nainstalován lokálně, nebo k němu přistupovat vzdáleně po síti. Každý objekt je potřeba pro připojení k DB nastavit a nakonfigurovat podle druhu databáze.

Pro použití v simulačním modelu jsou tedy k dispozici předpřipravené tyto objekty pro připojení k databázím:

- SQLite,
- ODBC,
- Oracle11g,
- Oracle19c.

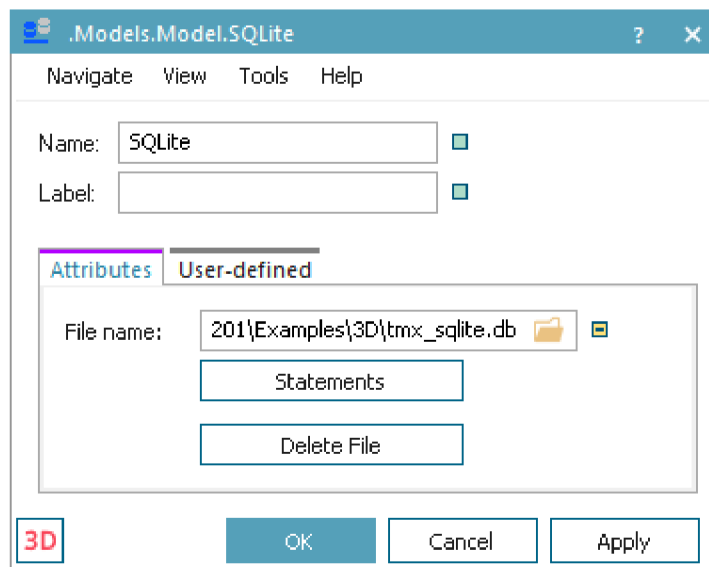
3.1 SQLite

Objekt SQLite je softwarová knihovna, která implementuje samostatný, bezserverový, transakční databázový SQL stroj, který implementuje většinu standardu SQL92 a který je součástí Plant Simulation. [7]

Jelikož DB SQLite ukládá strukturu a data do souborů, typicky na lokálním disku a každá databáze tak má svůj vlastní soubor, zásadním nastavením pro připojení k DB SQLite je nastavení cesty k databázovému souboru. Pro případ užití v této práci se soubor nacházel v počítači na lokálním úložišti, ale obecně je možno nastavit libovolnou funkční cestu na jakýkoli disk.

Minimálním nastavením pro SQLite DB (Obr. 3.1) je tedy nastavení parametrů:

- název objektu,
- cesta k databázovému souboru.



Obr. 3.1 Nastavení DB SQLite - vlastní zpracování

3.2 Open Database Connectivity

Oproti rozhraním specifickým pro nějaký konkrétní produkt je výhodou ODBC to, že umožňuje přístup k různým zdrojům dat, jako mohou být např. MS Excel, DBASE, textové soubory, jakož i DBMS, jako Oracle, nebo MS SQL Server, pokud jsou v počítači uživatele nainstalovány příslušné ovladače.

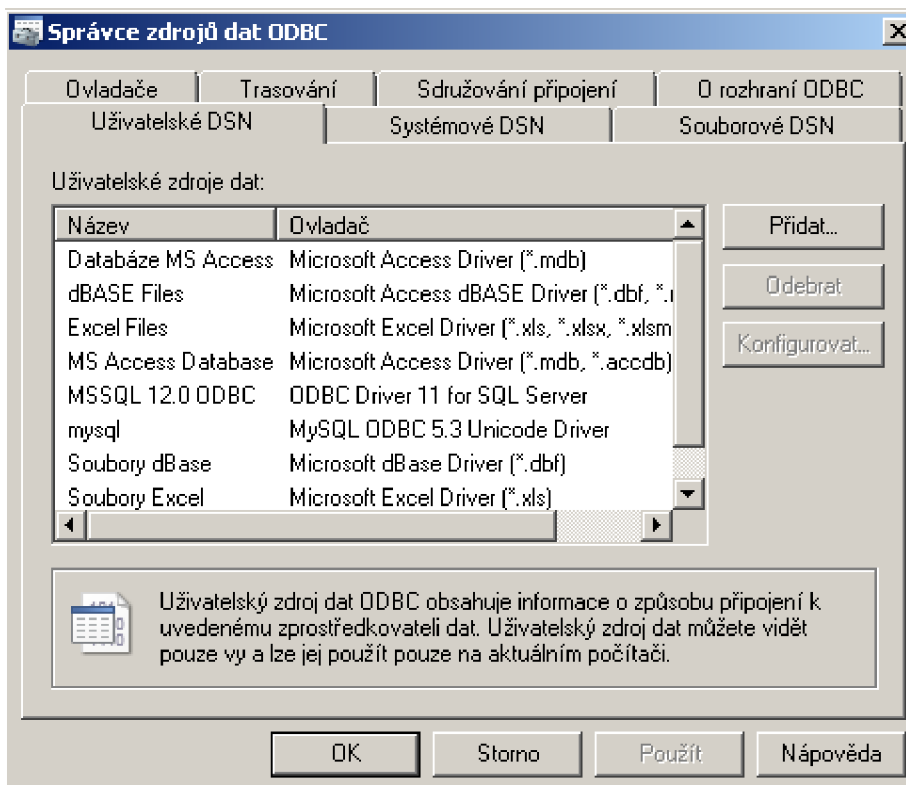
Pomocí objektu ODBC je v PtS možno se připojit k jakémukoli zdroji dat. Pro každé připojení k databázi se do simulačního modelu vloží objekt ODBC pro připojení k databázi. MS Windows poskytuje správce ovladačů ODBC, umožňuje načíst a spravovat ovladač potřebný pro konkrétní úkoly.

Jednoúrovňové ovladače jsou určeny pro zdroje dat, které nemají rozhraní SQL, jako jsou Microsoft Excel, DBASE, nebo textové soubory. V tomto případě ovladač zpracovává volání ODBC a příkazy SQL a převádí je na operace základních souborů. Tyto ovladače obvykle omezují nabízené příkazy SQL. Musí však poskytovat aspoň minimální příkazy definované ODBC.

Víceúrovňové ovladače vyžadují server, který zpracovává příkazy SQL a je rozhraním ke zdroji dat. Ovladač zpracovává volání ODBC a předává příkazy SQL na server. Je

také možné definovat volání specifické pro DBMS. Ovladač je napsán tak, aby odpovídal syntaxi požadované DBMS. Víceúrovňové ovladače jsou určeny pro typické architektury klient / server, jako jsou Oracle, MS SQL Server, a další. [7]

Pro každé jednotlivé připojení přes ODBC je do uživatelova počítače potřeba doinstalovat ODBC ovladače. Na počítači s MS Windows to je potřeba v případě použití DBMS jiného dodavatele, a někdy i v případě, že je na počítači lokálně instalován MS SQL Server je potřeba podle příslušné verze SQL Serveru doinstalovat potřebné ovladače (Obr. 3.2).



Obr. 3.2 Dialog MS Windows pro nastavení zdrojů ODBC - vlastní zpracování

Pro účely této práce byly vybrány 2 zdroje pro připojení přes ODBC:

- síťová databáze MariaDB běžící na lokálním webovém serveru Apache, standardně používaná jako webová databáze,
- MS SQL Server Express spouštěn z lokální instalace na PC.

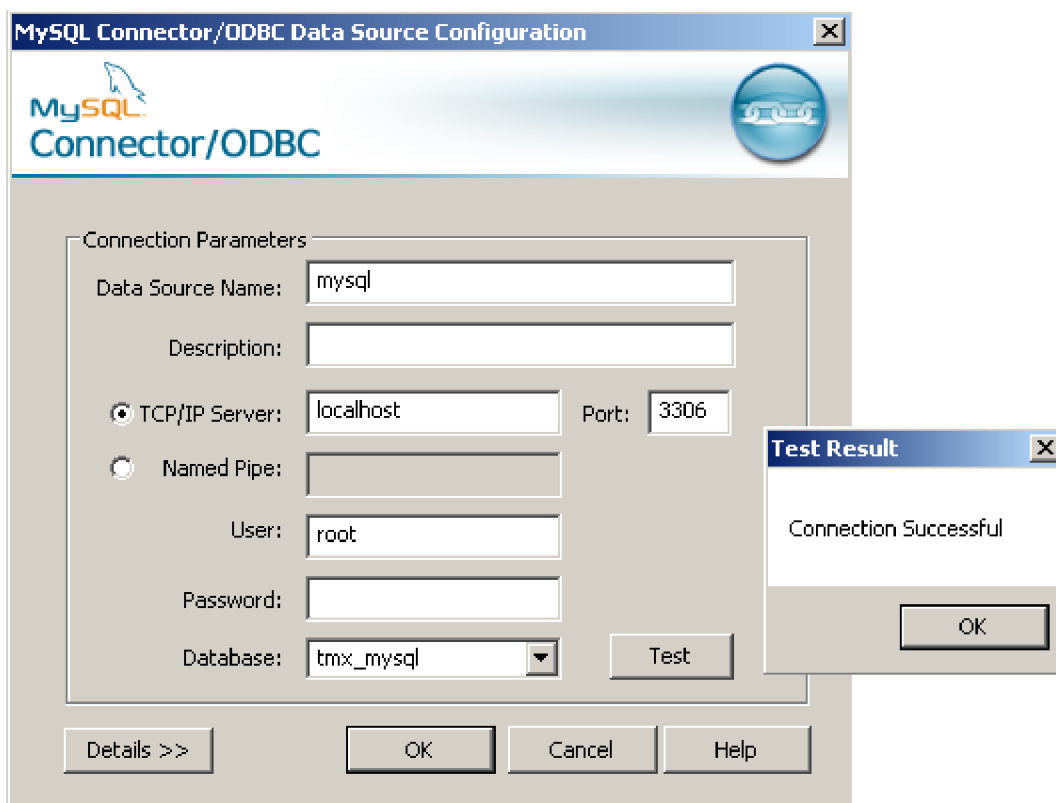
3.2.1 Nastavení připojení MySQL / MariaDB přes ODBC

Pro nastavení připojení MySQL / MariaDB přes ODBC je do počítače potřeba doinstalovat ovladač příslušné verze a provést nastavení připojení.

V nastavení je potřeba zadat:

- název zdroje dat,
- adresu serveru,
- port na kterém je databáze připojena,
- název uživatele databáze (dle DBMS, někdy nemusí být vyžadováno),
- heslo uživatele databáze (dle DBMS, někdy nemusí být vyžadováno),
- název databáze.

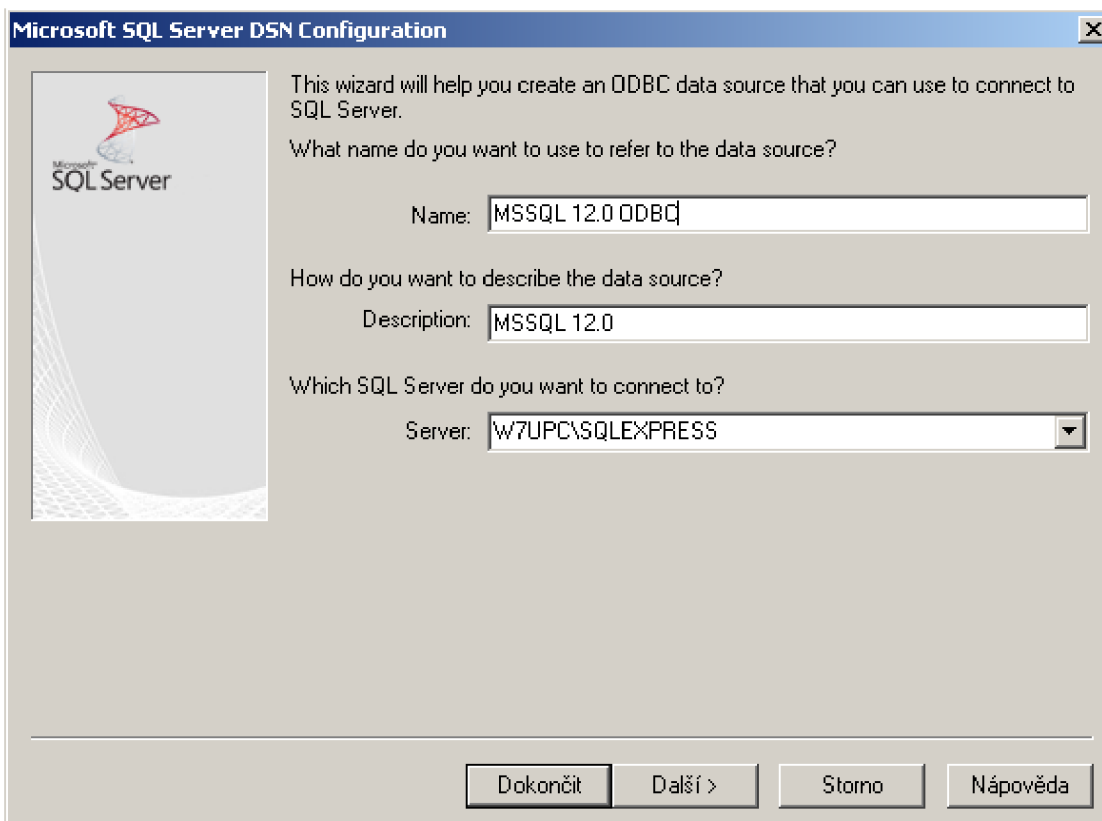
Po nastavení parametrů a po stisknutí tlačítka „Test“ proběhne test připojení k databázi (Obr. 3.3).



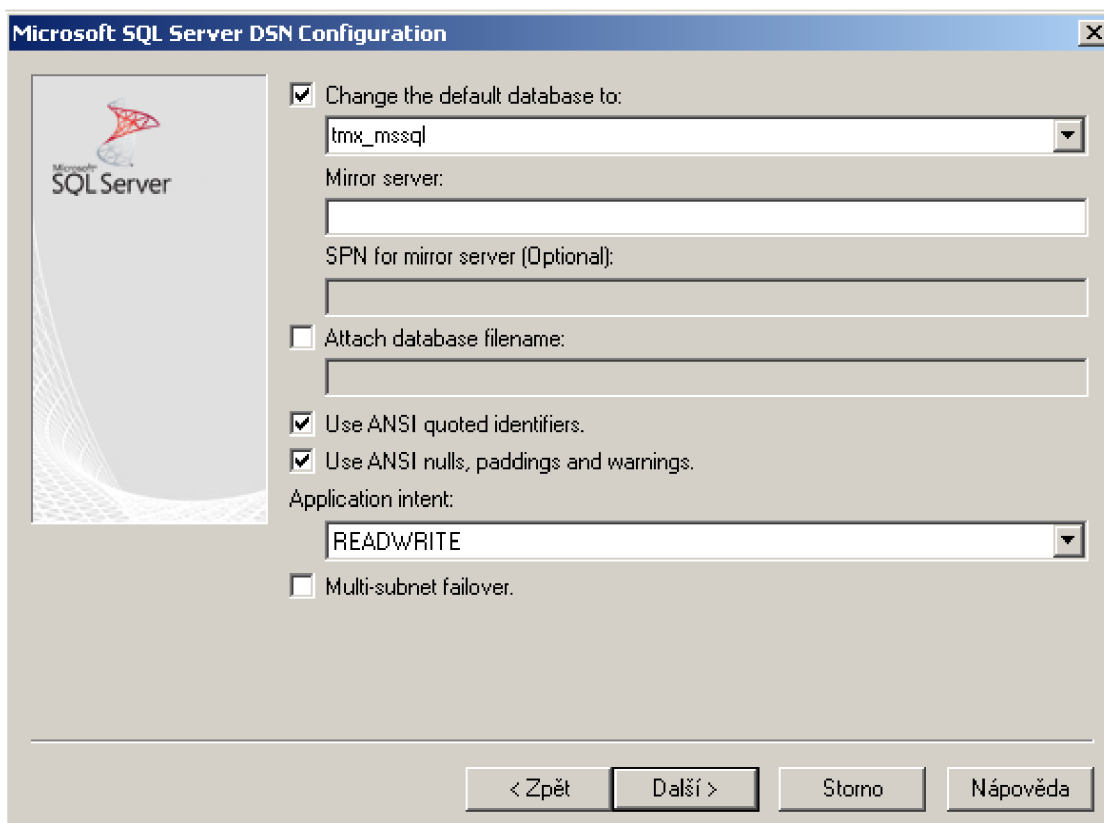
Obr. 3.3 Nastavení připojení MySQL / MariaDB přes ODBC - vlastní zpracování

3.2.2 Nastavení připojení MS SQL Server Express přes ODBC

Nastavení ODBC připojení pro MS SQL Server Express má více kroků, všemi uživatele provede průvodce nastavením, stačí zadat hodnoty a potvrdit volby (Obr 3.4. a 3.5).

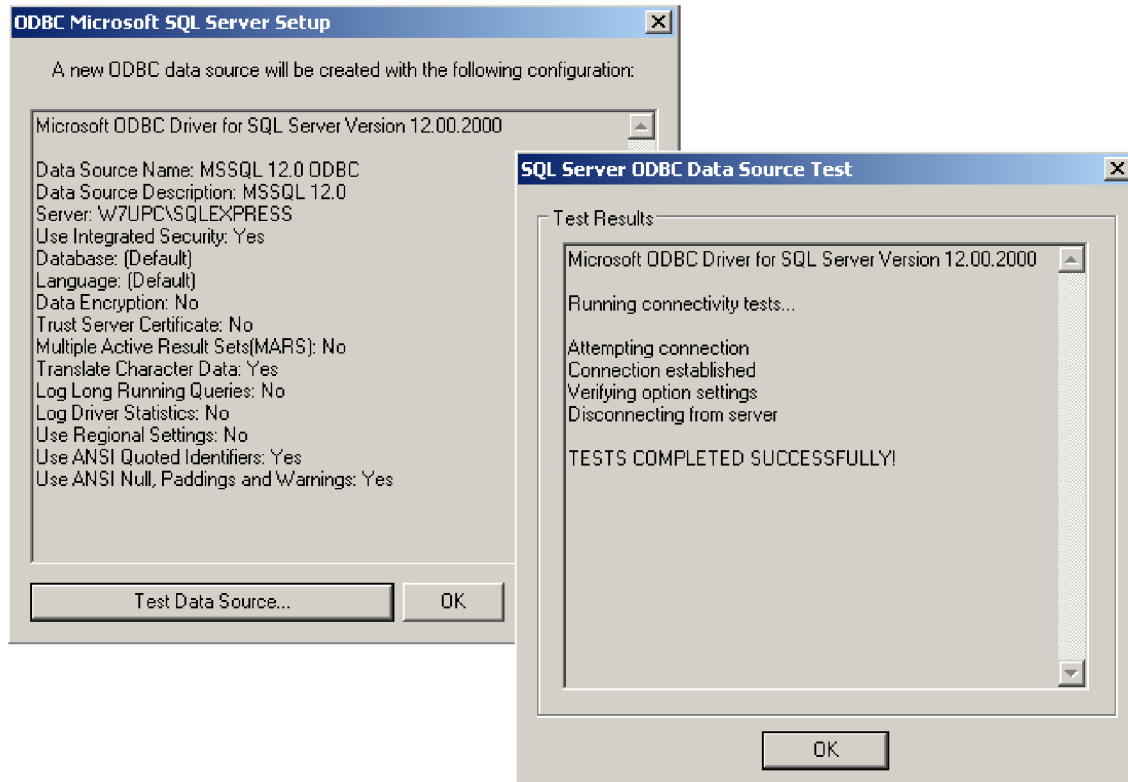


Obr. 3.4 Nastavení připojení MS SQL Server přes ODBC - vlastní zpracování



Obr 3.5 Výběr MS SQL Server databáze – vlastní zpracování

Po nastavení databáze a výběru defaultní databáze pro ukládání dat je možno ověřit, zda je připojení funkční (Obr. 3.6).



Obr. 3.6 Nastavení připojení MS SQL Server přes ODBC - vlastní zpracování

3.3 Oracle11g

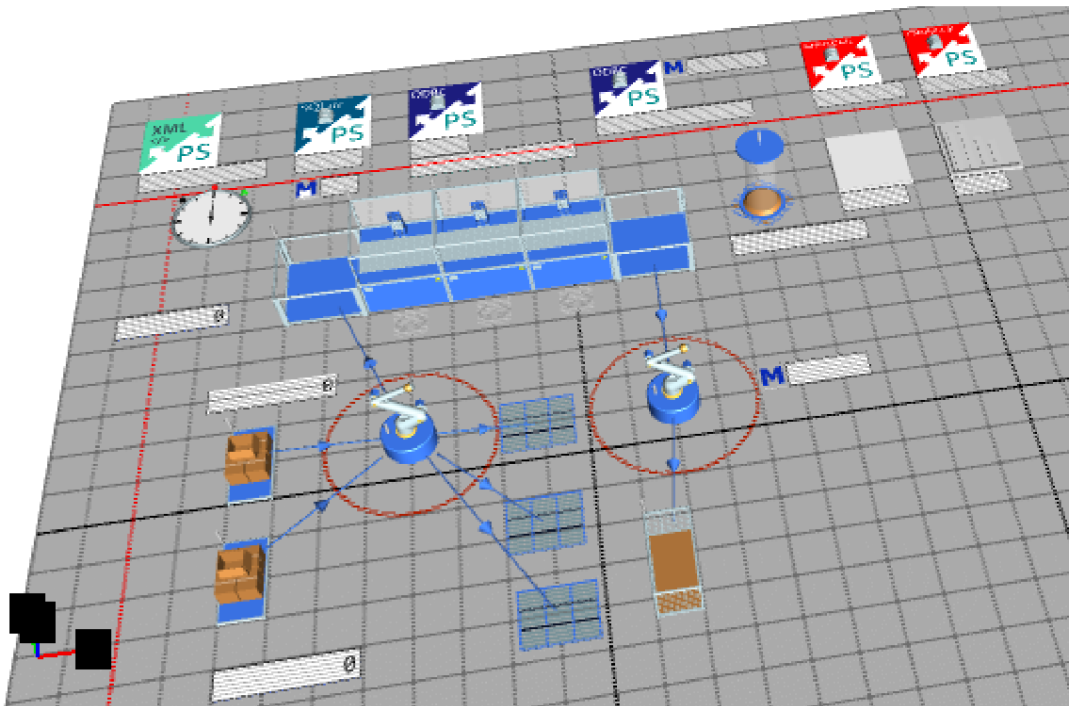
Objekt Oracle11g je určen pro nastavení a navázání spojení s externí databází Oracle. Tento objekt může navázat spojení s jakýmkoli databázovým serverem Oracle v síti, ke kterému má počítač s nainstalovaným PtS přístup. Je možno vytvořit maximálně šest současných připojení k různým databázovým serverům. Pro takovéto připojení se do modelu vloží samostatný objekt Oracle11g pro každé připojení v knihovně tříd. [7]

3.4 Oracle19c

Objekt Oracle19c slouží k připojení pro příslušnou vyšší verzi DB Oracle. Objekt pro připojení verzí Oracle 12c a Oracle 18c není v nabídce objektů PtS, ale je předpoklad, že se k nim lze připojit některým z uvedených objektů.

4 Ukázkové příklady

Po připojení k databázi je možno ji začít využívat. Pro případ této práce byl navržen jednoduchý simulační model, sestávající z výrobní linky o 3 pracovištích, vstupního a výstupního bufferu, 2 nezávislých vstupních zdrojů, 2 robotických manipulátorů, 3 mezioperačních odkládacích ploch a 1 výstupní odkládacího shromaždiště (Obr. 4.1). Model simuluje průchod vstupních polotovarů jednoduchou výrobní linkou, jejich zpracování na 3 dílčích operacích a jejich ukládání na výsledném úložišti.



Obr. 4.1 Simulační model výrobní linky - vlastní zpracování

Pro možnost využití databází byly přidány další samostatné prvky a objekty, které mohou být ovládány programovým kódem interního jazyka SimTalk.

Databázové objekty:

- objekt *SQLite* pro připojení k DB SQLite,
- objekt *ODBC* pro připojení k DB MySQL / MariaDB,
- objekt *ODBC* pro připojení k DB MS SQL Server,
- objekt *Oracle11g* pro připojení k DB Oracle,
- objekt *Oracle19c* pro připojení k DB Oracle.

Ostatní objekty:

- 3x Objekt *Variable* pro ukládání stavu proměnných,
- 3x samostatný „virtuální“ objekt *Method* pro záznam zdrojových kódů SimTalk.

Další objekty *Method* pro záznam zdrojových kódů SimTalk jsou přidružené k příslušným „fyzickým“ objektům.

2 z výše uvedených objektů *Variable*, *Iter01* a *Iter02* jsou přidruženy ke zdrojovým objektům *Zdroj01* a *Zdroj02* a ukládají okamžitou hodnotu stavu počítadla polotovarů, které zdroje předávají robotickému manipulátoru. Tyto hodnoty jsou následně pomocí programového kódu jazyka SimTalk ukládány do různých databází pro možnost dalšího zpracování externím informačním systémem, který může mít k těmto databázím přístup jako k univerzálnímu zdroji dat a mít je k dispozici současně s jinými potřebnými daty v databázích.

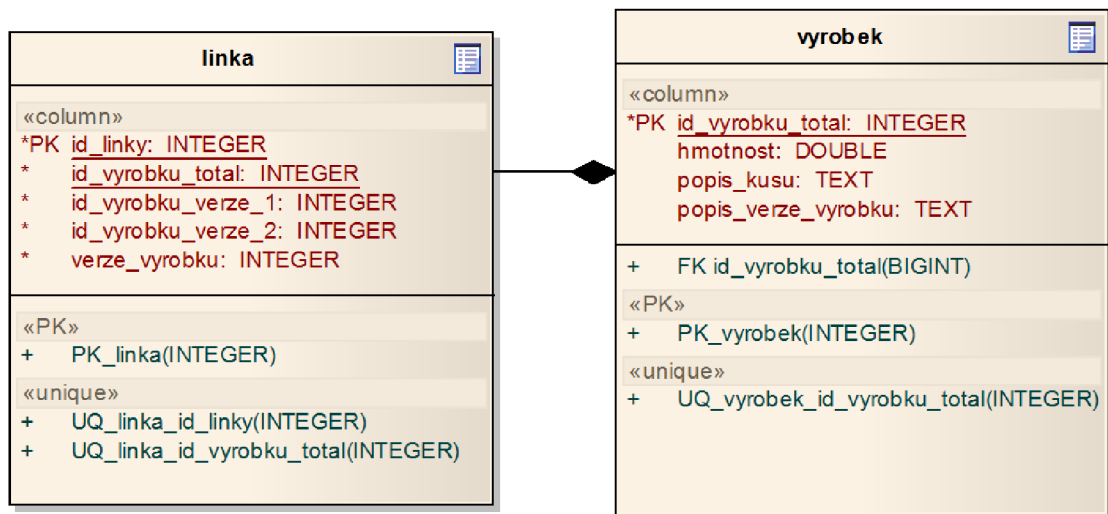
Objekt *Method* s názvem *Model.Models.Init* obsahuje uživatelsky definovaný programový kód jazyka SimTalk, který se v modelu spustí současně se spuštěním procesu simulace. Tento programový kód vykonává pro každou databázi stejnou funkčnost, ale je pro každou databázi jiný, podle různých požadavků na konkrétní syntaxi.

Programový kód SimTalk vykonává tyto databázové operace:

- otevře připojení k databázi,
- nastaví proměnnou s textovým řetězcem SQL příkazů pro:
 - vytvoření tabulky *linka* s několika sloupci, v případě, že tato neexistuje,
 - vytvoření tabulky *vyrobek* s několika sloupci, v případě, že tato neexistuje,
- spustí SQL příkazy,
- uzavře spojení s databází.

Dále model obsahuje objekty *Method*, které patří k „fyzickým“ objektům *Zdroj01* a *Zdroj02* a které obsahují programový kód SimTalk uživatelsky definovaných metod *Models.Model.Zdroj01.exitControl01* a *Models.Model.Zdroj02.exitControl02*. Objekty *Zdroj01* a *Zdroj02* předávají těmto metodám stav proměnných *Iter01* a *Iter02* a uvedené metody zapisují stav proměnných do databází.

Součástí výše uvedených databázových operací je vytvoření tabulek v databázi dle níže uvedeného schématu. V některých databázích se mohou lišit datové typy, např. datový typ *VARCHAR* může být nahrazen datovým typem *TEXT*, podle použitého druhu databáze. Databáze obsahuje 2 jednoduché tabulky - *linka* a *vyrobek*, tabulky mohou být propojeny přes atribut *id_vyrobku_total*, který je primárním klíčem obou tabulek a zároveň cizím klíčem v tabulce *vyrobek*. Kardinalita relace mezi tabulkami *linka* a *vyrobek* je 1:N (Obr. 4.2).



Obr. - 4.2. Schéma tabulek v databázi - vlastní zpracování

Tabulka *linka* je použita v této práci, tabulka *vyrobek* slouží jako ukázka možnosti rozšíření databázového schématu a mohla by být použita pro další práci s databází.

Tabulka *linka* obsahuje atributy:

- *id_linky* – primární klíč tabulky, datový typ INTEGER, NOT NULL, č. linky,
- *id_vyrobku_total* – datový typ INTEGER, NOT NULL, absolutní ID výrobku,
- *id_vyrobku_verze_1* – datový typ INTEGER, NOT NULL, ID výrobku verze,
- *id_vyrobku_verze_2* – datový typ INTEGER, NOT NULL, ID výrobku verze,
- *verze_vyrobku* – datový typ INTEGER, NOT NULL, rozlišovací znak verze výrobku.

Tabulka *vyrobek* obsahuje atributy:

- *id_vyrobku_total* – datový typ INTEGER, NOT NULL, ID výrobku, primární klíč tabulky a cizí klíč pro propojení tabulek,
- *hmotnost* – datový typ DOUBLE, hodnota hmotnosti výrobku,
- *popis_kusu* – datový typ VARCHAR (nebo TEXT), nepovinný parametr,
- *popis_verze_vyrobku* - datový typ VARCHAR (nebo TEXT), nepovinný parametr,
- *verze_vyrobku* – datový typ INTEGER, NOT NULL, rozlišovací znak verze výrobku.

4.1 Plant Simulation, SimTalk a interní DB SQLite

DB SQLite zapisuje strukturu a data databáze do lokálního souboru *tmx_sqlite.db*. Pro SQLite byly použity níže uvedené zápisy programového kódu.

4.1.1 Programový kód pro vytvoření tabulek, metoda *Model.Models.Init*

Součástí programového kódu SimTalk metody *Model.Models.Init* jsou příkazy SQL, které jsou v kódu programu vkládány do proměnné s názvem *sql* jako textový řetězec, který je odeslán databázi a ta tyto příkazy provede.

```
var sql:string
sqlite.open
sql:=""
"CREATE TABLE IF NOT EXISTS linka (id_linky INTEGER, "+
"id_vyrobku_total INTEGER PRIMARY KEY AUTOINCREMENT,
id_vyrobku_verze_1 INTEGER, "+
"id_vyrobku_verze_2 INTEGER, verze_vyrobku INTEGER);" +
"CREATE TABLE IF NOT EXISTS vyrobek (id_vyrobku_total
INTEGER PRIMARY KEY, "+
"hmotnost DOUBLE, popis_kusu TEXT, popis_verze_vyrobku
TEXT);"
sqlite.exec(sql)
sqlite.close
```

4.1.2 Programový kód pro zápis proměnných *Iter01* a *Iter02*

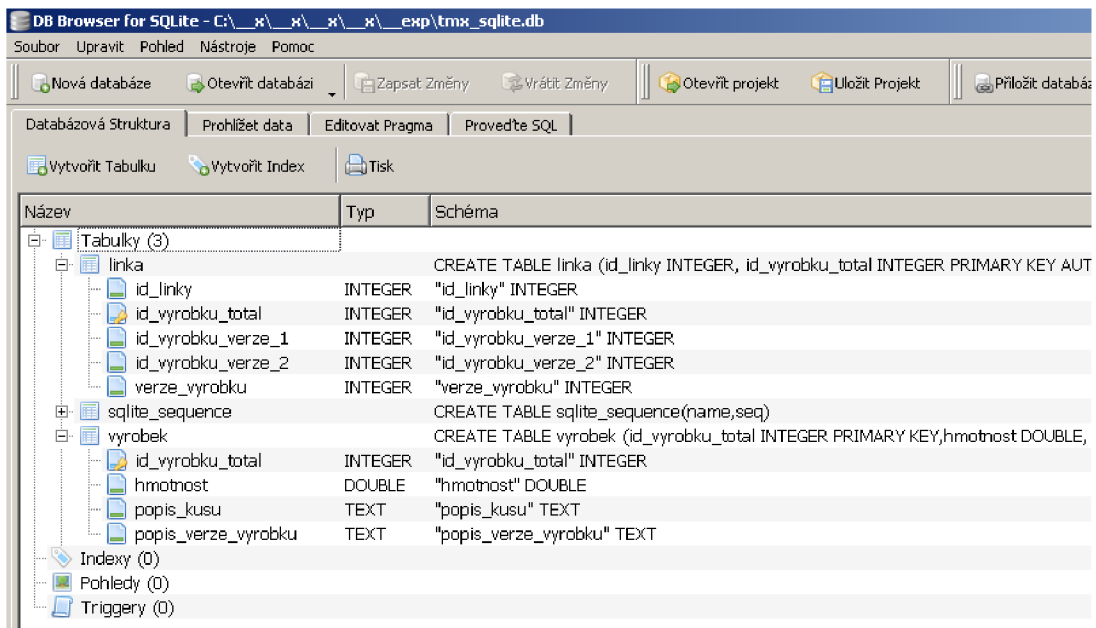
Metody *Models.Model.Zdroj01.exitControl01* a *Models.Model.Zdroj02.exitControl02* jsou stejné, liší se pouze proměnnými, které zapisují do DB. Konstantní parametry *id_vyrobku_verze_1* a *id_vyrobku_verze_2* slouží k rozlišení zdroje, z kterého polotovar vstupuje do linky. Je tak také možno např. rozlišovat atributy, které má ten který vstupní polotovar. SQL příkazy jsou opět vloženy do proměnných *sql01* a *sql02* jako textový řetězec a poté databázi vykonány.

```
Iter01:=Iter01+1;
var sql01:string
sqlite.open
sql01:="INSERT into linka (id_linky, id_vyrobku_verze_1,
verze_vyrobku) VALUES (1, "+Iter01+", 1)";
sqlite.exec(sql01)
sqlite.close
```

```
Iter02:=Iter02+1;
var sql02:string
sqlite.open
sql02:="INSERT into linka (id_linky, id_vyrobku_verze_2,
verze_vyrobku) VALUES (1, "+Iter02+", 2)";
sqlite.exec(sql02)
sqlite.close
```

4.1.3 Zobrazení databáze *tmx_sqlite.db* v DB Browser for SQLite

V níže uvedených obrázcích je zobrazena výsledná struktura SQLite databáze (Obr. 4.3) vytvořená programovým kódem SimTalk a data vložená do SQLite databáze po spuštění modelu v Plant Simulation. Po spuštění modelu jsou tedy v případě, že dosud neexistují, vytvořeny 2 tabulky a poté do nich simulační model zapisuje svá data generovaná procesem běhu linky (Obr. 4.4).



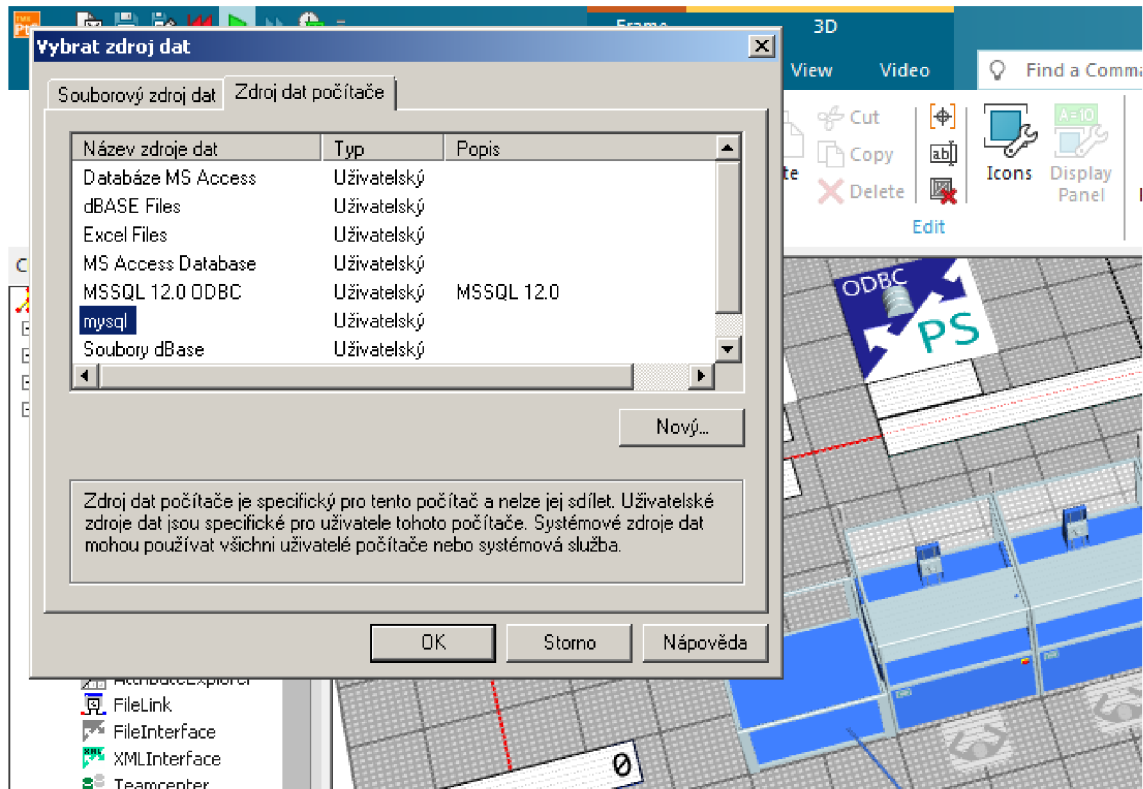
Obr. 4.3 SQLite databázová struktura – vlastní zpracování

	id_linky	id_vyrobtu_total	id_vyrobtu_verze_1	id_vyrobtu_verze_2	verze_vyrobtu
	Filtr	Filtr	Filtr	Filtr	Filtr
1	1	1	1	NULL	1
2	1	2	NULL	1	2
3	1	3	2	NULL	1
4	1	4	3	NULL	1
5	1	5	NULL	2	2
6	1	6	4	NULL	1
7	1	7	5	NULL	1
8	1	8	6	NULL	1
9	1	9	NULL	3	2
10	1	10	7	NULL	1
11	1	11	8	NULL	1
12	1	12	9	NULL	1
13	1	13	NULL	4	2
14	1	14	10	NULL	1
15	1	15	11	NULL	1
16	1	16	12	NULL	1
17	1	17	NULL	5	2
18	1	18	13	NULL	1

Obr. 4.4 SQLite - uložená data - vlastní zpracování

4.2 Plant Simulation, SimTalk a připojení přes ODBC k externí síťové databázi MariaBD

Jako síťová (používaná převážně jako webová) databáze pro připojení přes ODBC (Obr. 4.5) byla vybrána databáze MariaDB (vývojová odnož MySQL), pro případ této práce běžící na lokálním PC jako součást balíku XAMPP. Balík XAMPP slouží pro simulaci webového serveru Apache pro možnost vývoje webových aplikací na lokálním PC.



Obr. 4.5 Výběr připojení ODBC - vlastní zpracování

Ale připojení k databázi by bylo možné i ke vzdálenému místu po síti. Jako DBMS bylo použito HeidiSQL. Níže uvedené zápisy programového kódu SimTalk jsou velmi podobné předešlému příkladu, obsahují ale drobné odchylky v syntaxi potřebné pro připojení před ODBC. Další drobné změny v syntaxi SQL pak souvisí s různými zápisy příkazů jazyka SQL pro různé databáze.

4.2.1 Programový kód pro vytvoření tabulek, metoda *Model.Models.Init*

Podobně jako v předešlém případě programový kód SimTalk metody *Model.Models.Init* použije příkazy SQL vložené do proměnné jako textový řetězec, který je odeslán

databázi a ta tyto příkazy provede. Z porovnání obou příkladů vyplynou drobné rozdíly v syntaxi SimTalk a SQL pro ODBC připojení a databázi MariaDB. V tomto případě byly pro zápis SQL použity 2 proměnné z důvodu lepší přehlednosti zdrojového kódu. Kód pro přihlášení k DB neobsahuje povinné parametry *název DB*, *login* a *heslo*, neboť tyto parametry jsou předávány již v nastavení ODBC. Zde záleží na volbě uživatele.

```
var sql_odbc1:string
var sql_odbc2:string

mysql.login("", "", "") -- parametry nastaveny v ODBC
sql_odbc1:="" +
"CREATE TABLE IF NOT EXISTS linka (id_linky INTEGER, "+
"id_vyrobku_total INTEGER AUTO_INCREMENT,
id_vyrobku_verze_1 INTEGER, "+
"id_vyrobku_verze_2 INTEGER, verze_vyrobku INTEGER, PRIMARY
KEY (id_vyrobku_total));"
sql_odbc2:="" +
"CREATE TABLE IF NOT EXISTS vyrobek (id_vyrobku_total
INTEGER AUTO_INCREMENT, "+
"hmotnost DOUBLE, popis_kusu TEXT, popis_verze_vyrobku
TEXT, PRIMARY KEY(id_vyrobku_total));"

mysql.sql(sql_odbc1)mysql.sql(sql_odbc2)
mysql.logout
```

4.2.2 Programový kód pro zápis proměnných *Iter01* a *Iter02*

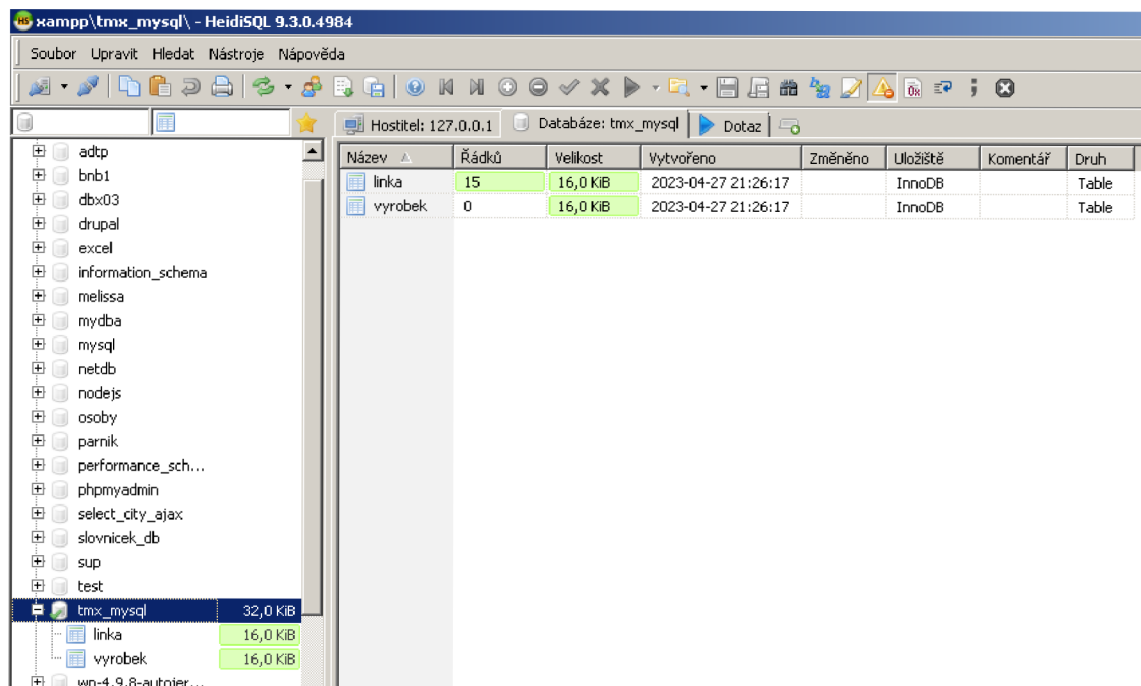
Metody *Models.Model.Zdroj01.exitControl01* a *Models.Model.Zdroj02.exitControl02* jsou podobně jako v předešlém příkladu opět stejné a liší se pouze proměnnými, které zapisují do DB. Konstantní parametry *id_vyrobku_verze_1* a *id_vyrobku_verze_2* slouží k rozlišení zdroje, z kterého polotovar vstupuje do linky. Je tak také možno např. rozlišovat atributy, které má ten který vstupní polotovar. SQL příkazy jsou vloženy do proměnných *sql01* a *sql02* jako textový řetězec a poté databázi vykonány (Obr. 4.6). Z porovnání příkladů opět vyplynou drobné rozdíly v syntaxi SimTalk a SQL pro ODBC připojení a databázi MariaDB.

```

Iter01:=Iter01+1;
var sql01:string
mysql.login("", "", "") -- parametry nastaveny v ODBC
sql01:="INSERT into linka (id_linky, id_vyrobku_verze_1,
verze_vyrobku) VALUES (1, "+Iter01+", 1)";
mysql.sql(sql01)
mysql.logout

Iter02:=Iter02+1;
var sql02:string
mysql.login("", "", "") -- parametry nastaveny v ODBC
sql02:="INSERT into linka (id_linky, id_vyrobku_verze_2,
verze_vyrobku) VALUES (1, "+Iter02+", 2)";
mysql.sql(sql02)
mysql.logout

```



Obr. 4.6 Vytvořené tabulky *linka* a *vyrobek* - vlastní zpracování

Na obrázcích níže je uvedena struktura vytvořených tabulek (Obr. 4.7 a 4.8).

Hostitel: 127.0.0.1 | Databáze: tmx_mysql | Tabulka: linka

Název: linka

Komentář:

Sloupce:

#	Název	Datový typ	Délka/Množina	Unsigned	Nulový	Zerofill	Výchozí
1	id_linky	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
2	id_vyrobu_total	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
3	id_vyrobu_verze_1	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	id_vyrobu_verze_2	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
5	verze_vyrobu	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Obr. 4.7 Struktura tabulky *linka* - vlastní zpracování

Hostitel: 127.0.0.1 | Databáze: tmx_mysql | Tabulka: vyrobek

Název: vyrobek

Komentář:

Sloupce:

#	Název	Datový typ	Délka/...	Unsigned	Nulový	Zerofill	Výchozí	Kom
1	id_vyrobu_total	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	
2	hmotnost	DOUBLE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
3	popis_kusu	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Žádná hodnota	
4	popis_verze_vyrobu	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Žádná hodnota	

Obr. 4.8 Struktura tabulky *vyrobek* - vlastní zpracování

Po spuštění simulačního modelu se do databáze začínají ukládat předávané parametry (Obr. 4.9).

id_linky	id_vyrobku_total	id_vyrobku_verze_1	id_vyrobku_verze_2	verze_vyrobku
1	1	1	(NULL)	1
1	2	(NULL)	1	2
1	3	2	(NULL)	1
1	4	3	(NULL)	1
1	5	(NULL)	2	2
1	6	4	(NULL)	1
1	7	5	(NULL)	1
1	8	6	(NULL)	1
1	9	(NULL)	3	2
1	10	7	(NULL)	1
1	11	8	(NULL)	1
1	12	9	(NULL)	1
1	13	(NULL)	4	2
1	14	10	(NULL)	1
1	15	11	(NULL)	1
1	16	12	(NULL)	1
1	17	(NULL)	5	2
1	18	13	(NULL)	1
1	19	14	(NULL)	1
1	20	15	(NULL)	1
1	21	(NULL)	6	2

Obr. 4.9 Data vložená do tabulky *linka* spuštěním simulace v modelu - vlastní zpracování

4.3 Plant Simulation, SimTalk a připojení přes ODBC k externí lokální databázi MS SQL Server Express

MS SQL Server patří k nejrozšířenějším databázovým systémům a proto je vhodné se jím v této práci zabývat. Mnoho firem je uživatelem právě nějaké verze MS SQL Serveru.

Pro případ této práce běží MS SQL Server Express na lokálním PC. Jako DBMS bylo použito Microsoft SQL Server Management Studio, které se instaluje jako součást MS SQL Serveru.

Příklady zápisů programového kódu SimTalk jsou opět podobné předešlým příkladům, obsahují ale drobné odchylky v syntaxi potřebné pro připojení před ODBC a další odlišnosti v syntaxi SQL, které souvisí se syntaxí jazyka T-SQL, který je používán pro MS SQL Server.

4.3.1 Programový kód pro vytvoření tabulek, metoda *Model.Models.Init*

Podobně jako v předešlých případech programový kód SimTalk metody *Model.Models.Init* použije příkazy SQL vložené do proměnné jako textový řetězec, který je odeslán databázi a ta tyto příkazy provede. Z porovnání příkladů vyplynou drobné rozdíly v syntaxi SimTalk a SQL pro ODBC připojení a databázi MS SQL Server. V tomto případě byly pro zápis SQL použity 2 proměnné z důvodu lepší přehlednosti zdrojového kódu. Kód pro přihlášení k DB neobsahuje povinné parametry *název DB*, *login* a *heslo*, neboť tyto parametry jsou předávány již v nastavení ODBC a při lokálním použití MS SQL Serveru je zde navíc možnost použít autentifikaci pomocí Windows Authentication, součásti OS Windows.

```
var sql_odbc1:string
var sql_odbc2:string
-- v případě lokální instalace MSSQL Server Express
přihlášení přes Windows Authentication
mssql.login("", "", "")
-- při práci s MS SQL Serverem je nutno použít příkazy
jazyka T-SQL
sql_odbc1:="" +
"IF NOT EXISTS (SELECT * FROM sysobjects WHERE NAME='linka'
AND xtype='U') "+
"CREATE TABLE linka (" +
"  id_linky INT, "+
"  id_vyrobku_total INT IDENTITY(1,1) PRIMARY KEY NOT NULL, "+
"  id_vyrobku_verze_1 INT, "+
"  id_vyrobku_verze_2 INT, "+
"  verze_vyrobku INT);"
-- GO (v MS SQL Server Management Studio potřeba zadat na
konci sady příkazů větve IF)

sql_odbc2:="" +
"IF NOT EXISTS (SELECT * FROM sysobjects WHERE
NAME='vyrobek' AND xtype='U') "+
```

```
"CREATE TABLE vyrobek ("
" id_vyrobku_total INT IDENTITY(1,1) PRIMARY KEY NOT NULL,"
" hmotnost DECIMAL(6,2),"
" popis_kusu TEXT,"
" popis_verze_vyrobku TEXT);"
```

```
mssql.sql(sql_odbc1)
mssql.sql(sql_odbc2)
mssql.logout
```

4.3.2 Programový kód pro zápis proměnných *Iter01* a *Iter02*

Metody *Models.Model.Zdroj01.exitControl01* a *Models.Model.Zdroj02.exitControl02* jsou podobně jako v předešlých příkladech opět stejné a liší se pouze proměnnými, které zapisují do DB. Konstantní parametry *id_vyrobku_verze_1* a *id_vyrobku_verze_2* slouží k rozlišení zdroje, z kterého polotovár vstupuje do linky. Je tak také možno např. rozlišovat atributy, které má ten který vstupní polotovár. SQL příkazy jsou vloženy do proměnných *sql01* a *sql02* jako textový řetězec a poté databázi vykonány (Obr. 4.10). Z porovnání příkladů opět vyplynou drobné rozdíly v syntaxi SimTalk a SQL pro ODBC připojení a databázi MS SQL Server.

```
inSequ01:=insequ01+1;
var sql01:string

mssql.login("", "", "")
sql01:="INSERT into linka (id_linky, id_vyrobku_verze_1,
verze_vyrobku) VALUES (1, "+inSequ01+", 1)";
mssql.sql(sql01)
mssql.logout
inSequ02:=insequ02+1;
var sql02:string
mssql.login("", "", "")
sql02:="INSERT into linka (id_linky, id_vyrobku_verze_2,
verze_vyrobku) VALUES (1, "+inSequ02+", 2)";
mssql.sql(sql02)
mssql.logout
```

tmx_mssql.sql - W7UPC\SQLSERVER.tmx_mssql (w7upc\w7u (53))* - Microsoft SQL Server Management Studio (Administrátor)

Object Explorer

W7UPC\SQLSERVER (SQL Server 12.0.2000 - w7upc\w7u)

tmx_mssql

Database Diagrams

Tables

System Tables

FileTables

dbo.linka

Columns

- id_linky (int, null)
- id_vyrobku_total (PK, int, not null)
- id_vyrobku_verze_1 (int, null)
- id_vyrobku_verze_2 (int, null)
- verze_vyrobku (int, null)

Keys

Constraints

Triggers

Indexes

Statistics

dbo.vyrobek

Columns

- id_vyrobku_total (PK, int, not null)
- hmotnost (decimal(6,2), null)
- popis_kusu (text, null)
- popis_verze_vyrobku (text, null)

Keys

Constraints

Triggers

Indexes

Statistics

Views

Synonyms

Programmability

Service Broker

Storage

Security

```

19 SELECT * FROM [tmx_mssql].[dbo].[linka];
20
21 -- WHERE (verze_vyrobku = 2) AND (id_vyrobku_total > 9)
22 -- ORDER BY id_vyrobku_verze_2 ASC;

```

	id_linky	id_vyrobku_total	id_vyrobku_verze_1	id_vyrobku_verze_2	verze_vyrobku
1	1	1	1	NULL	1
2	1	2	NULL	1	2
3	1	3	2	NULL	1
4	1	4	3	NULL	1
5	1	5	NULL	2	2
6	1	6	4	NULL	1
7	1	7	5	NULL	1
8	1	8	6	NULL	1
9	1	9	NULL	3	2
10	1	10	7	NULL	1
11	1	11	8	NULL	1
12	1	12	9	NULL	1
13	1	13	NULL	4	2
14	1	14	10	NULL	1
15	1	15	11	NULL	1
16	1	16	12	NULL	1
17	1	17	NULL	5	2
18	1	18	13	NULL	1
19	1	19	14	NULL	1
20	1	20	15	NULL	1
21	1	21	NULL	6	2
22	1	22	16	NULL	1
23	1	23	17	NULL	1
24	1	24	18	NULL	1
25	1	25	NULL	7	2
26	1	26	19	NULL	1
27	1	27	20	NULL	1

Query executed successfully. W7UPC\SQLSERVER (12.0 RTM) | w7upc\w7u (53)

Výstup

Zobrazit výstup z: Object Explorer

```

(tbl.name=vyrobek and SCHEMA_NAME(tbl.schema_id)=dbo)
ORDER BY
clms.column_id ASC

```

Obr. 4.10 Ukázka struktury tabulek *linka* a *vyrobek* vytvořené příkazem programového kódu SimTalk v DB MS SQL Server Express a dat vložených do tabulky *linka* spuštěním simulace v modelu - vlastní zpracování

Data uložená do databáze pak lze třídit, či jinak zpracovávat běžným způsobem pomocí SQL příkazů nad databází. V ukázce (Obr. 4.11) je uveden příklad selekce dat získaných z ukázkového simulačního modelu podle vybraného atributu.

The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'tmx_mssql'. The central query window contains the following SQL code:

```

19 SELECT * FROM [tmx_mssql].[dbo].[linka]
20 WHERE (verze_vyrobku = 2) AND (id_vyrobku_total > 9)
21 ORDER BY id_vyrobku_verze_2 ASC;
22

```

The Results pane shows the following data table:

	id_linky	id_vyrobku_total	id_vyrobku_verze_1	id_vyrobku_verze_2	verze_vyrobku
1	1	13	NULL	4	2
2	1	17	NULL	5	2
3	1	21	NULL	6	2
4	1	25	NULL	7	2
5	1	29	NULL	8	2
6	1	33	NULL	9	2

Obr. 4.11 Třídění dat uložených v DB pomocí SQL příkazů – vlastní zpracování

Závěr

V této práci byly testovány možnosti připojení Tecnomatix Plant Simulation k různým databázím.

Z hlediska řešené problematiky nebylo možno otestovat Plant Simulation databázové objekty pro připojení Oracle11g a Oracle19c. I k těmto databázím by bylo možno se připojit přes ODBC, ale Plant Simulation pro Oracle nabízí samostatné speciální objekty. Jistě by bylo také vhodné lépe navrhnout strukturu tabulek podle pravidel pro databázové normální formy a více dalších vylepšení. To vše by mohlo být tématem nějaké případné budoucí závěrečné práce.

V uváděných příkladech bylo zjištěno, že všechny testované databáze lze při správném nastavení připojit bez problémů a proměnné simulovaného procesu z modelu byly v reálném čase předávány a ukládány do databází. Takto uložená data by mohla být dále zpracovávána jakoukoli aplikací, která se může připojit k relační databázi, nebo může být databáze použita jako úložiště dat v případě, kdy by bylo potřeba z modelu exportovat velké objemy dat.

V práci je také uveden ukázkový příklad simulačního modelu v Tecnomatix Plant Simulation, který generuje výstupní data, a pomocí interního jazyka SimTalk je ukládá do různých databází. Tento postup by mohl být vhodný v případě, kdy je např. požadováno mít k dispozici velké objemy dat vyexportovány do relační databáze mimo samotný Plant Simulation, ať už kvůli jejich objemu, kvůli potřebě je zpracovávat jiným softwarem, nebo jen jako externí zálohu pro Plant Simulation.

Tato data jsou pak k dispozici samostatně a uživatel je může dále zpracovávat nezávisle na Plant Simulation.

Vzhledem k tomu, že se v blízké budoucnosti očekává zpracování velkých objemů dat, ať už v souvislosti s AI, nebo v souvislosti s potřebami průmyslu 4.0, je zřejmé, že ukládání velkých objemů dat ze software pro simulaci výrobních procesů do externích databází by mohlo mít smysl. I když tzv. Big data mohou používat ve větší míře již nestrukturovaná data, pro uložení dat v tzv. datovém skladu, jako ve zvláštním typu relační databáze, mohou být data ukládána ze simulačního modelu procesu do externí relační databáze jistě vhodnou formou vstupu. Takto uložená data se také mohou uplatnit jako zdroj dat pro tzv. digitální dvojče.

Seznam zdrojů

- [1] BANGSOW, Steffen. Manufacturing Simulation with Plant Simulation and SimTalk Usage and Programming with Examples and Solutions. Springer-Verlag Berlin Heidelberg, 2010. 297 s. ISBN-13: 978-3-642-05073-2.
- [2] BANGSOW, Steffen. Tecnomatix Plant Simulation Modeling and Programming by Means of Examples. Second Edition. Springer Nature Switzerland AG, 2020. ISBN-13: 978-3-030-41544-0.
- [3] TREBUŇA, Peter - MARKOVIČ, Jaromír - KLIMENT, Marek - HALČÍNOVÁ, Jana. Modelovanie v priemyselnom inžinierstve. Prvé vydanie. Košice, TU v Košiciach, Strojnícka fakulta, 2015. 195 s. Edícia vedeckej a odbornej literatúry. ISBN-13: 978-80-553-1953-7.
- [4] Tecnomatix. [online]. Siemens PLM Software. Nedatováno [cit. 2022-12-18] Dostupné z: www.plm.automation.siemens.com/global/en/products/tecnomatix
- [5] CVETKOVIĆ, Dragan. Computer Simulation. London, IntechOpen Limited, 2017. 264 s. ISBN-13: 978-9535132059.
- [6] ZEIGLER, Bernard P. – MUZY, Alexandre – KOFMAN, Ernesto. Theory of Modeling and Simulation, Discrete Event and Iterative System Computational Foundations. Third Edition. London, Elsevier Inc. (Academic Press), 2019. 667 s. ISBN-13: 978-0-12-813370-5.
- [7] Tecnomatix Plant Simulation Help [online]. Nedatováno [cit. 2023-04-19] Dostupné z: https://docs.plm.automation.siemens.com/content/plant_sim_help/15/plant_sim_all_in_one_html/en_US/tecnomatix_plant_simulation_help/simtalk_reference/simtalk_reference.html
- [8] Microsoft Office Access [online]. Nedatováno [cit. 2023-04-19] Dostupné z: www.microsoft.com/cs-cz/microsoft-365/access
- [9] Apache OpenOffice [online]. Nedatováno [cit. 2023-04-19] Dostupné z: www.openoffice.org
- [10] HeidiSQL [online]. Nedatováno [cit. 2023-04-19] Dostupné z: www.heidisql.com
- [11] DB Browser for SQLite [online]. Nedatováno [cit. 2023-04-19] Dostupné z: <https://sqlitebrowser.org>
- [12] phpMyAdmin [online]. Nedatováno [cit. 2023-04-19] Dostupné z: www.phpmyadmin.net
- [13] Microsoft SQL Server Management Studio [online]. Nedatováno [cit. 2023-04-19] Dostupné z: <https://learn.microsoft.com/en-us/sql/ssms>

[14] Microsoft SQL Server [online]. Nedatováno [cit. 2023-04-19]
Dostupné z: www.microsoft.com/en-us/sql-server

[15] Oracle Database [online]. Nedatováno [cit. 2023-04-19]
Dostupné z: www.oracle.com/database

Seznam grafických objektů

Seznam obrázků

Obr. 2.1 Ukázka simulace procesu v Tecnomatix Plant Simulation [4].....	21
Obr. 2.2 Pokus o připojení k souboru MSO Access z LibreOffice přes ODBC driver pro MSO Access – vlastní zpracování	23
Obr. 2.3 Test připojení k databázi MariaDB z LibreOffice přes ODBC driver pro MySQL – vlastní zpracování	24
Obr. 2.4 Pokusný dotaz na databázi MariaDB z LibreOffice - vlastní zpracování	24
Obr. 2.5 Prostředí aplikace HeidiSQL - vlastní zpracování	25
Obr. 2.6 Prostředí aplikace XAMPP - vlastní zpracování	26
Obr. 2.7 Prostředí aplikace phpMyAdmin a připojení k databázi MariaDB - vlastní zpracování	27
Obr. 2.8 SQL Server Configuration Manager, spuštění SQL Serveru - vlastní zpracování	27
Obr. 2.9 SQL Server Configuration Manager, spuštění SQL Serveru - vlastní zpracování	28
Obr. 2.10 Microsoft SQL Server Management Studio, připojení k SQL Serveru - vlastní zpracování	28
Obr. 2.11 Microsoft SQL Server Management Studio, připojení k SQL Serveru - vlastní zpracování	29
Obr. 2.12 Prostředí aplikace Oracle SQL Developer a připojení k databázi - vlastní zpracování	30
Obr. 2.13 Oracle SQL Developer dotaz na databázi Oracle - vlastní zpracování	30
Obr. 3.1 Nastavení DB SQLite - vlastní zpracování	35
Obr. 3.2 Dialog MS Windows pro nastavení zdrojů ODBC - vlastní zpracování	36
Obr. 3.3 Nastavení připojení MySQL / MariaDB přes ODBC - vlastní zpracování	37
Obr. 3.4 Nastavení připojení MS SQL Server přes ODBC - vlastní zpracování	38
Obr. 3.5 Výběr MS SQL Server databáze – vlastní zpracování	38
Obr. 3.6 Nastavení připojení MS SQL Server přes ODBC – vlastní zpracování	39

Obr. 4.1 Simulační model výrobní linky - vlastní zpracování	40
Obr. 4.2 Schéma tabulek v databázi - vlastní zpracování	42
Obr. 4.3 SQLite databázová struktura – vlastní zpracování	45
Obr. 4.4 SQLite - uložená data - vlastní zpracování	45
Obr. 4.5 Výběr připojení ODBC - vlastní zpracování.....	46
Obr. 4.6 Vtvořené tabulky <i>linka</i> a <i>vyrobek</i> - vlastní zpracování.....	48
Obr. 4.7 Struktura tabulky <i>linka</i> - vlastní zpracování.....	49
Obr. 4.8 Struktura tabulky <i>vyrobek</i> - vlastní zpracování.....	49
Obr. 4.9 Data vložená do tabulky <i>linka</i> spuštěním simulace v modelu - vlastní zpracování.....	50
Obr. 4.10 Ukázka struktury tabulek <i>linka</i> a <i>vyrobek</i> vytvořené příkazem programového kódu SimTalk v DB MS SQL Server Express a dat vložených do tabulky <i>linka</i> spuštěním simulace v modelu - vlastní zpracování	53
Obr. 4.11 Třídění dat uložených v DB pomocí SQL příkazů – vlastní zpracování	54

Seznam zkratek

AI Artificial Intelligence (umělá inteligence)

API Application Programming Interface (rozhraní pro programování aplikací)

CRUD Create, Read, Update, Delete (operace vytvoření , čtení, editace a mazání např. záznamu v databázi)

DB Databáze

DBC Database Connectivity (připojení k databázi)

DBMS Database Management System (systém řízení báze dat, SŘBD)

DBS Database system (databázový systém)

DLL Dynamic Link Libraries

GUI Grafical User Interface (grafické uživatelské rozhraní)

MS Microsoft Corporation

ODBC Open Database Connectivity

OS Operační systém

SŘBD viz DBMS

SW software

T-SQL Transact-SQL (proprietární rozšíření jazyka SQL od společností Microsoft a Sybase, které Microsoft používá v produktu Microsoft SQL Server)

Siemens PLM Software Siemens Product Lifecycle Management Software Inc.

UI User Interface (uživatelské rozhraní)

SW software

Autor BP	Martin Chytka
Název BP	Tecnomatix Plant Simulation - využití databází a programování v SimTalk
Studijní program	Logistika / Informatika pro logistiku
Rok obhajoby BP	2023
Počet stran	55
Počet příloh	0
Vedoucí BP	prof. Ing. Gabriel Fedorko, PhD.
Anotace	<p>Současný trend přechodu na digitální a automatizovanou výrobu, obecně označován jako Průmysl 4.0, vyžaduje rozsáhlé změny v přístupu k návrhu produktů, technologií, výrobních linek a mnoha dalších záležitostí s takovouto výrobou souvisejících s takovouto výrobou souvisejícího. Tlak na zkracování času mezi návrhem výrobku, prototypem a výrobou mj. zvyšuje požadavky na automatizaci opakovaných rutinních postupů při práci obecně s každým softwarem, kde to bude možné a vhodné. Cílem této bakalářské práce je ověřit možnosti připojení simulačního software Tecnomatix Plant Simulation k různým databázím a navrhnout případové studie využití databází a programování v SimTalk k automatizaci rutinních činností v Plant Simulation.</p>
Klíčová slova	Logistika, výroba, simulace procesů, Tecnomatix Plant Simulation, databáze, SimTalk
Místo uložení	ITC (knihovna) Vysoké školy logistiky v Přerově
Signatura	