

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Diplomová práce**

**Databázový systém knihy jízd v podniku**

**Bc. Petra Andrejsková**

© 2018 ČZU v Praze

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Petra Andrejsková

Projektové řízení

Název práce

**Databázový systém knihy jízd v podniku**

Název anglicky

**Database system of log book in the company**

---

### Cíle práce

Diplomová práce je zaměřena na relačně databázové řešení problematiky knihy jízd v podnikatelském subjektu. Hlavním smyslem a účelem této závěrečné práce bude:

- a) objasnit teoretické principy relačně databázové technologie v kontextu s jejím využitím v problematice evidence knihy jízd v malých a středních podnicích,
- b) zmapovat a dokumentovat momentální stav této problematiky na konkrétním příkladu a vymezit její relevantnost včetně požadavků na ni kladených,
- c) navrhnout akceptovatelné řešení této záležitosti s přihlédnutím ke splnění výše vymezených požadavků,
- d) demonstrovat a ověřit vlastní funkčnost navrženého řešení,
- e) ověřené záležitosti zobecnit pro další možná uplatnění.

### Metodika

Použitá metodika zadané diplomové práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Stěžejními metodami této práce budou metody a techniky relačně databázové technologie a SQL. Navrhované řešení bude zohledňovat identifikované požadavky a očekávání spojená s řešenou záležitostí. Na podkladě syntézy teoretických poznatků a dosažených výsledků budou formulovány závěry této diplomové práce a následně zobecněny pro další možná použití.

Závazný harmonogram:

Teoretické principy řešené problematiky, literární rešerše – předmět 1. zápočtu z DP: do 30.5.2018

Zmapování momentální situace řešené problematiky, identifikace požadavků s tím spojených: do 31.7.2018

Navržení možného řešení a jejich následné ověření: předmět 2. zápočtu z DP: do 15.10.2018

Zobecnění navržených záležitostí pro další možná použití – předmět 3. zápočtu z DP: do 15.11.2018

## Doporučený rozsah práce

55-65

## Klíčová slova

relačně databázová technologie, SQL, kniha jízd, datové modelování

---

## Doporučené zdroje informací

CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK, 2009. Mistrovství – databáze: profesionální průvodce tvorbou efektivních databází. Brno: Computer Press. ISBN 978-802-5123-287.

MCLAUGHLIN, Michael a John M. HARPER, 2015. Oracle Database 12c PL/SQL advanced programming techniques. New York: McGraw-Hill Education. ISBN 978-007-1835-145.

PRICE, Jason., 2014. Oracle database 12c SQL. New York: McGraw-Hill. ISBN 978-007-1799-355.

URMAN, Scott, Ron HARDMAN a Michael MCLAUGHLIN, 2008. Oracle: programování v PL/SQL. Brno: Computer Press. Programování (Computer Press). ISBN 978-802-5118-702.

VALENTA, M. – POKORNÝ, J. *Databázové systémy*. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9.

---

## Předběžný termín obhajoby

2018/19 ZS – PEF (únor 2019)

## Vedoucí práce

doc. Dr. Ing. Václav Vostrovský

## Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 14. 11. 2018

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 14. 11. 2018

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 18. 11. 2018

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci „Databázový systém knihy jízd v podniku“ jsem vypracovala samostatně pod vedením vedoucího diplomové 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 autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 29. listopadu 2018

---

### **Poděkování**

Děkuji doc. Ing. Václavu Vostrovskému, Ph.D., za odborné konzultace, poskytované cenné rady a připomínky, kterými přispěl k vytvoření této diplomové práce.

Děkuji společnosti VUSTE-APIS, s. r. o., jmenovitě Ing. Josefu Vančurovi, CSc., generálním řediteli a jednatelem společnosti, Ing. Martinu Andreasovi, výkonnému řediteli, Ing. Vojtěchu Vančurovi, řediteli produkce ICT, za důvěru, trpělivost, rady a za vytvoření vlídného prostředí pro realizaci zadané práce.

Děkuji mé matce Mgr. Janě Andrejskové a Bc. Evě Heribanové, DiS., za všestrannou podporu a dále Ing. Janu Lobovskému, MBA, za překlad do anglického jazyka.

# Databázový systém knihy jízd v podniku

## Abstrakt

Diplomová práce je zaměřena na relačně databázové řešení problematiky knihy jízd v podnikatelském subjektu. Hlavním smyslem a účelem této závěrečné práce bude:

- a) objasnit teoretické principy relačně databázové technologie v kontextu s jejím využitím v problematice evidence knihy jízd v malých a středních podnicích,
- b) zmapovat a dokumentovat momentální stav této problematiky na konkrétním příkladu a vymezit její relevantnost včetně požadavků na ni kladených,
- c) navrhnout akceptovatelné řešení této záležitosti s přihlédnutím ke splnění výše vymezených požadavků,
- d) demonstrovat a ověřit vlastní funkčnost navrženého řešení,
- e) ověřené záležitosti zobecnit pro další možná uplatnění.

Použitá metodika zadané diplomové práce bude založena na studiu a analýze dostupných informačních zdrojů a existujících řešení v dané oblasti. Stěžejními metodami této práce budou metody a techniky relačně databázové technologie SQL a PL/SQL. Navrhované řešení bude zohledňovat identifikované požadavky a očekávání spojená s řešenou záležitostí. Na podkladě syntézy teoretických poznatků a dosažených výsledků budou formulovány závěry této diplomové práce a následně zobecněny pro další možná použití.

**Klíčová slova:** relačně databázová technologie, SQL, kniha jízd, datové modelování

# **Database system of log book in the company**

## **Abstract**

This Diploma Thesis focuses on the relational database solution of a Database system of log book in the company. Main target of this Diploma Thesis is to:

- a) Clarify the underlying principles of relational Database technology applied into Database system of log book in the company.
- b) Map and document the current status of this subject matter in terms of a specific example and define the subject matter requirements.
- c) To design the acceptable solution of this subject matter, solution that is build based upon the given requirements (re b))
- d) Demonstrate and verify the functionality of the designed solution
- e) Summarize the functionality verification and outline its future possible usability

Applied methodology of this Diploma Thesis was based upon the study and analysis of the available information resources and existing field solutions. The essential methods of this Thesis are the techniques related to the relational database technologies SQL and PL/SQL. The designed solution matches the requirements and exception criteria defined prior to the solution as such. The conclusions of this Diploma Theses are based on the synthesis of the relational database theory and achieved practical results. Last but not least, these results will be summarized in terms of possible future practical Journey Log utilization.

**Keywords: Relational Databases, SQL, Log Book, Data Modelling**

# Obsah

<b>1 Úvod.....</b>	<b>11</b>
<b>2 Cíl práce a metodika .....</b>	<b>14</b>
2.1 Cíl práce .....	14
2.2 Metodika .....	14
<b>3 Teoretická východiska .....</b>	<b>16</b>
3.1 Databáze .....	16
3.1.1 Data vs. informace .....	16
3.1.2 SŘBD .....	16
3.1.3 Databáze.....	16
3.1.4 Databázový systém .....	17
3.2 Databázové modely .....	18
3.2.1 Hierarchický model.....	18
3.2.2 Síťový model .....	18
3.2.3 Relační model .....	18
3.3 Další důležité pojmy .....	20
3.3.1 Entita .....	20
3.3.2 Atribut .....	20
3.3.3 Klíče Relace .....	21
3.3.4 Primární klíč .....	21
3.3.5 Cizí klíč.....	21
3.3.6 Kardinalita .....	22
3.3.7 Pohled .....	23
3.3.8 Formulář.....	23
3.4 Databázový jazyk SQL .....	23
3.4.1 Struktura jazyk SQL .....	24
3.4.2 DDL .....	25
3.4.3 DML.....	26
3.4.4 Transakční příkazy.....	26
3.5 Procedurální jazyk PL/SQL .....	27
3.5.1 Struktura příkazu PL/SQL .....	27
3.5.2 Funkce a procedury .....	28
3.5.3 Trigger .....	28
3.5.4 Sekvence .....	29
3.5.5 Proměnné .....	30
3.5.6 Řídící struktury .....	31
3.5.6.1 Podmínky IF a CASE .....	31



3.5.6.2	Příkaz LOOP .....	31
3.5.7	Kurzor .....	32
3.6	Oracle SQL Developer .....	33
3.7	Report .....	33
3.7.1	Jaspersoft Studio .....	33
<b>4</b>	<b>Vlastní práce .....</b>	<b>34</b>
4.1	Vymezení problému .....	34
4.2	Framework .....	34
4.3	Požadavky na nový systém .....	35
4.4	Analýza a návrh .....	38
4.4.1	Logický návrh databáze .....	38
4.4.2	Datový slovník .....	38
4.4.3	Návrh funkcí formulářů .....	40
4.5	Implementace .....	41
4.5.1	Tvorba tabulek .....	41
4.5.2	Vizuální podoba formulářů .....	44
4.5.3	Programování .....	48
4.5.3.1	Procedura CONTROL_CAR .....	48
4.5.3.2	Procedura LOAD_DATE .....	49
4.5.3.3	Procedura KEY_COMMIT .....	50
4.5.3.4	Procedury BUTT_NEW a BUTT_DEL .....	51
4.5.3.5	Procedura LOG_USE .....	52
4.5.3.6	Procedura DETAIL .....	53
4.5.4	Report .....	54
4.5.5	Uživatelská příručka .....	58
4.5.5.1	Obecné ovládaní .....	58
4.5.5.2	Formuláře .....	58
4.5.5.3	Založení nového záznamu .....	60
<b>5</b>	<b>Výsledky a diskuze .....</b>	<b>62</b>
<b>6</b>	<b>Závěr .....</b>	<b>65</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>68</b>
<b>8</b>	<b>Přílohy .....</b>	<b>70</b>
<b>Seznam obrázků</b>		
Obrázek 1 -	Databázový systém .....	17
Obrázek 2 -	Entita, atribut, relace .....	20
Obrázek 3 -	Primární a cizí klíč .....	22

Obrázek 4 - Struktura jazyka SQL.....	24
Obrázek 5 - Příkaz vytvoření Triggeru .....	29
Obrázek 6 - Příkaz sekvence pro Oracle databázi .....	30
Obrázek 7 - Třívrstvá frameworková architektura .....	35
Obrázek 8 - MS Excel s požadavky na zadávané parametry .....	37
Obrázek 9 - Logický model .....	38
Obrázek 10 - Tabulka VX_CL_LOG .....	42
Obrázek 11 - Pohled QVX_CL_LOG_OVERVIEW .....	43
Obrázek 12 - Tabulka CODEBOOK .....	44
Obrázek 13 - Zobrazení knihy jízd v menu .....	45
Obrázek 14 - Přehledový formulář F_P_VX_CL_CARS.....	45
Obrázek 15 - Editační formulář F_E_VX_CL_CARS .....	46
Obrázek 16 - Přehledový formulář F_P_VX_CL_LOG.....	46
Obrázek 17 - Editační formulář F_E_VX_CL_LOG .....	47
Obrázek 18 - Přehledový formulář F_P_VX_CL_OVERVIEW .....	48
Obrázek 19 - Připojení k databázi.....	54
Obrázek 20 - Grafický návrh reportu knihy jízd.....	57
Obrázek 21 - Panel nástrojů.....	58
Obrázek 22 - Upozornění.....	59
Obrázek 23 - Přehledový formulář .....	59
Obrázek 24 - Editační formulář .....	60
Obrázek 25 - Založení nového záznamu.....	61

## Seznam tabulek

Tabulka 1 - Datový slovník .....	39
----------------------------------	----

## Seznam použitých zkratk

DB .....	DataBase
DBMS .....	DataBase Management System
DBS.....	DataBase System
DDL .....	Data Definitoin Language
DML.....	Data Manipulation Language
ID .....	IDentification
IS .....	Informační systém
JDBC.....	Java Database Connectivity
PL/SQL .....	Procedural Language/Structured Query Language
SQL.....	Structured Qeury Language
SŘBD.....	Systém řízení báze dat
VAMAX .....	VA Manager
VUSTE.....	Výzkumný ústav strojírenské technologie a ekonomiky
APIS.....	Agentura poradenství a inženýrských služeb

# 1 Úvod

Diplomová práce „Databázový systém knihy jízd v podniku“ řeší problematiku knihy jízd v podnikatelském subjektu, a to konkrétně ve společnosti VUSTE-APIS, s. r. o. Tato společnost se zabývá vývojem a implementací softwaru, nabízí softwarové produkty a specializuje se na servis softwarových aplikací. Tyto služby poskytuje státním organizacím a průmyslovým podnikům, zaměřuje se na veřejný sektor v České republice, ale i na Slovensku.

Uvedené téma diplomové práce si autorka zvolila především proto, že databázový systém jízd prakticky realizovala ve výše zmíněné firmě jako svůj první zadaný samostatný projekt, neboť zde poslední dva roky studia byla zaměstnána na částečný úvazek. Vznikla tak efektivně využívaná funkční část systému v podnikové struktuře, konkrétně v podnikovém systému VAMAX. Výsledek její práce je nyní využíván všemi zaměstnanci uvedené společnosti.

V současné době problematika, kterou se zabývá tato diplomová práce, jde ruku v ruce s velice aktuálním rozvojem informačních technologií v podnikovém prostředí a s efektivním využitím času ve firmě. Z údajů Eurostatu (Evropského statistického úřadu) vyplývá, že Češi za hodinu vyrobí zhruba jen 71 % toho, co za stejnou dobu zvládne průměrný zaměstnanec Evropské unie. Paradoxem je, že Češi pracují v průměru déle než například Němci.

Nízká produktivita práce je mnohdy způsobena složitě nastavenými procesy uvnitř firmy a jejich neefektivním zvládnutím. Podniky se nesoustředí na firemní prioritu, mnoho času tráví rutinními činnostmi, velká část prací je řešena neefektivně a manuálně. Firmy většinou nevyužívají moderní nástroje, které nabízí dnešní informační technologie. Moderní aplikace umožňují předávání důležitých informací a nevyžadují zvláštní technické vybavení, stačí jen připojení k internetu nebo intranetu. Faktem je, že technologický vývoj je některými firmami stále podceňován. V elektronické podobě lze v dnešní době standardně spravovat nejen faktury, cestovní příkazy, docházku, výkazy práce, ale např. i knihu jízd.

Knihy jízd je vlastně účetní doklad, kde jsou zaznamenány údaje o jednotlivých jízdách vozidla v obchodním majetku či soukromého vozu používaného k podnikání. Knihu jízd potřebuje každá firma nebo OSVČ, která vlastní alespoň jedno vozidlo a využívá ho k výkonu své podnikatelské činnosti, a to ze dvou důvodů: první hledisko je finanční, např. možnosti odpočtu daní, a druhým důvodem je bezpečnost práce. Žádný zákon sice

nenarizuje knihu jízd vést, ale pro finanční úřad je třeba doložit kolik vozidlo, registrováno jako firemní, najelo celkem kilometrů za rok a kolik kilometrů z toho celku bylo vedeno jako služební cesta. Že je kniha jízd potřebná, dokládá i statistika nově registrovaných vozidel pro rok 2017. Z celkového počtu 271 595 nově registrovaných osobních automobilů tvořilo 74,6 % firemních vozů a jen 25,4 % privátních (dle statistik uvedených na <https://krize15.cz>).

V mnoha podnicích, i ve společnosti VUSTE-APIS, s. r. o., prošla kniha jízd vývojem. Nejdříve byla vedena v papírové podobě a posléze v tabulce v Excelu. Obecně je doporučována elektronická kniha jízd, protože šetří práci i čas a údaje jsou připravené pro budoucí použití, mohou být kdykoliv vytištěné. Faktem je, že technologický vývoj je v této oblasti některými firmami neustále podceňován, přesto může významnou měrou zvýšit efektivitu práce, a ta je klíčem k úspěchu podniku v konkurenčním prostředí. Toto je jeden z důvodů, proč byla navržena a aktualizována kniha jízd ve firmě VUSTE-APIS, neboť se tato společnost zabývá softwarovými produkty. Uvedená firma používá vlastní informační systém VAMAX, který sama vyvinula, a využívá ho v intranetové i internetové komunikaci. VA manager, zkráceně VAMAX, je systémem řízení a administrace podnikových nebo projektových procesů. Je podnikovým a informačním systémem nové generace, který svou koncepcí i technologiemi zpracovává informace. Systém VAMAX prochází neustálým vývojem a dochází k vylepšování dosavadních verzí. Jeho třívrstvá architektura se zakládá na datech uložených v Oracle databázi. Na ni navazuje logická struktura a úroveň programována převážně v jazyce PL/SQL. Dále v poslední vrstvě je systém VAMAX zobrazen ve webovém prohlížeči.

VAMAX je bezpečný systém, používá vysoký technický standard zabezpečení, ke svému spuštění nepotřebuje žádný zvláštní specializovaný program, dá se spustit pomocí běžného internetového prohlížeče. Je nejen sdíleným nástrojem pro libovolně definovaný počet spolupracovníků, ale pomocí VA manageru je možné s kolegy přímo komunikovat. VAMAX je stavebnicí, která poskytuje právě tolik služeb, kolik uživatel potřebuje, a ten se může soustředit pouze na svou práci. Tento informační systém nové generace firmě šetří nejen čas, ale i peníze a přispívá tak k efektivitě práce.

Současná kniha jízd ve firmě VUSTE-APIS, s. r. o. využívá právě systém VAMAX. Předchozí varianta knih, tedy ve formě excelové tabulky, nebyla dostačující i proto, že byla vedena jako samotná jednotka mimo systém VAMAX. Každý zaměstnanec měl svoji vlastní šablonu v Excelu, kterou posílal finančnímu oddělení v podniku e-mailem koncem každého měsíce. Nebylo možné získat přehled všech jízd a k nim náležitosti, jako například: počet

ujetých kilometrů různých zaměstnanců v určitých měsících a za celý rok. V excelové variantě firma nemohla účinně sledovat náklady vynaložené na pohonné hmoty, v přehledném výkazu nemohla oddělit služebně a soukromě ujeté kilometry.

Nové řešení knihy jízd je zabudováno v systému jako součást pilíře administrace uživatelů. Nabízí víceúrovňový přístup k jednotlivým vozidlům v rámci knihy jízd. Zaměstnanec, který má přiřazeno vozidlo, má omezený přístup k zobrazení přehledů, zatímco finanční oddělení a vedení firmy má přístup ke všem záznamům s neomezenou možností úprav.

Nová kniha jízd dále umožňuje přehled o kompletním vozovém parku firmy a jejich jízdách na jednom místě v podnikovém systému VAMAX s možností filtrování dle potřeby. Některá pole mají pro snadnější a rychlejší vyplňování výběrový seznam s možností volby z předem daných údajů. Zjednodušení je také v hledání určité jízdy napříč všemi zaměstnanci. Uživatel snadno zadá konkrétní datum a přehledový formulář ihned zobrazí daný záznam. Výrazně tak snižuje administrativu při vedení knihy jízd.

Autorka si uvědomuje, že kniha jízd mohla být dále zpracovávána v excelové tabulce, ta ale neumožňovala grafické zpracování formulářů pro uživatelsky přívětivější vkládání dat do polí s možností vygenerování reportu ve formátu PDF. Inovovaná kniha jízd dále umožňuje přehled o jízdách všech zaměstnanců za celý rok na jednom místě. Velkou předností nové verze je vyplňování knihy jízd kdekoli a odkudkoliv, kde je internetové připojení.

Vytvořený prototyp byl testován před ostrým nasazením do provozu. V rámci testování se objevily problémy, a to zejména v nastavení fontu u jednotlivých polí v reportu. To způsobilo nesprávné vygenerování a následně nečitelnost reportu. Výše uvedený problém byl řešen změnou fontu z defaultně nastaveného Sans Serif na Arial Narrow. Řešení bylo úspěšné a chyba byla odstraněna. V současné době je kniha jízd v systému VAMAX bezproblémově využívána všemi zaměstnanci společnosti VUSTE-APIS, s. r. o.

## 2 Cíl práce a metodika

### 2.1 Cíl práce

Diplomová práce je zaměřena na tvorbu elektronické knihy jízd v databázovém systému Oracle a na její využití v rámci podnikatelského subjektu. **Hlavním cílem** této předkládané diplomové práce je realizace SW řešení funkční knihy jízd v podnikovém systému VAMAX, která povede ke snížení administrace při vedení knihy jízd a bude zajišťovat větší transparentnost této záležitosti včetně možnosti víceúrovňového systému pro přístup k jednotlivým záznamům. **Dílními stěžejními kroky** této SW realizace budou:

- požadavky na nový systém,
- navržení logického modelu databáze,
- tvorba formuláře prostřednictvím grafického prostředí pro uživatelsky snadnější zápis dat do databáze,
- zhotovené reportu generujícího data ve formátu PDF v takové formě, aby mohl být vytisknut ve formátu A4 a archivován v tištěné podobě,
- vytvoření písemné uživatelské příručky pro každý formulář.

Těchto pět stěžejních dílních cílů této diplomové práce je dále rozděleno na konkrétní kroky, které jsou shrnuty kapitole 2.2 Metodika.

V úvodní teoretické části diplomové práce bude uvedena odborná rešerše, která je důležitá pro porozumění a seznámení s principy relační databáze, jazyky SQL a PL/SQL a pro pochopení kontextu zvolené problematiky. V praktické části budou naplněny cíle a kroky vydefinované výše.

### 2.2 Metodika

Zvolená metodika bude obsahovat následujících kroky:

- vyhledávání relevantních informačních zdrojů vhodných k zvolené problematice,
- studium dostupných odborných informačních zdrojů v oblasti problematiky relační databázové technologie, programování v jazyku SQL a PL/SQL,
- analýzu a zmapování stávajícího řešení vedení knihy jízd v podniku VUSTE-APIS s. r. o.,
- analýzu náležitostí obsahu databáze a knihy jízd,
- sumarizaci teoretických poznatků a analýzu problému,

- navržení nového řešení, tvorbu logického modelu databáze, zejména pro jasné znázornění propojení jednotlivých tabulek a splnění referenční integrity, odstranění redundantních dat a splnění normalizace dat podle pravidel tzv. normálních forem,
- navržení datového slovníku pro jasnou specifikaci dat, zejména pro dodržení doménové a entitní integrity,
- programování a realizaci nového prototypu elektronické knihy jízd v databázovém prostředí od firmy Oracle s vizuálním propojením v podnikovém systému VAMAX za pomoci datového modelování,
- návrh a realizaci reportu s výstupním formátem PDF v aplikaci Jaspersoft Studio,
- napsání uživatelských příruček k příslušným formulářům,
- závěrečné výsledky a diskuzi nad řešením zvolené problematiky.

## 3 Teoretická východiska

### 3.1 Databáze

V této části bude nejprve ujasněno: co jsou to data, kde jsou uložena, jak jsou strukturována a jaký systém obsahuje a spravuje data, respektive databáze.

#### 3.1.1 Data vs. informace

Mezi daty a informacemi je nepatrný rozdíl. Data jsou fakta, z kterých jsou informace odvozeny, můžeme je vnímat jako stavební jednotky informací. Data jako takové jsou zřídka užitečná, pokud jsou samostatná. Aby se mohla stát informacemi je třeba je dát do kontextu. Informace jsou obvykle formovány takovým způsobem, aby jim rozuměl příjemce. Pro počítač je typické, že přestože čte data, neznamená to, že by je chápal. Teprve použitím vzorců, programovacích skriptů nebo softwarových aplikací může počítač přeměnit data na informace, kterým můžou rozumět i příjemce.

Data jsou uložena v databázích, kde se nachází v organizované a strukturované podobě. Přísně definované databáze najdeme v počítačových strukturách. Ty šetří, organizují, chrání a poskytují data. Systém, který obsahuje a spravuje databáze, se nazývá systém řízení báze dat neboli SŘBD. V publikacích se můžeme setkat i s anglickým pojmem DBMS – DataBase Management System.

#### 3.1.2 SŘBD

SŘBD neboli systém řízení báze dat. Tento název vznikl z anglického termínu DataBase Management System – DBMS. Jde o softwarový nástroj určený k definování, manipulaci, načítání a správě dat v databázi. Tzn. že tvoří rozhraní mezi aplikacemi a uloženými daty. SŘBD obecně zpracovává jak data samotná, tak formát dat, názvy polí, strukturu záznamů a souborů. Rovněž definuje pravidla pro ověření a manipulaci s těmito daty. Příklady SŘBD jsou Oracle Database, MySQL, DB2, MS SQL Server a další.

#### 3.1.3 Databáze

*„Databáze je úložiště údajů, které jsou uloženy a zpracovávány nezávisle na aplikačních programech. Databáze zapouzdřují vlastní údaje, ale i relační vztahy mezi*



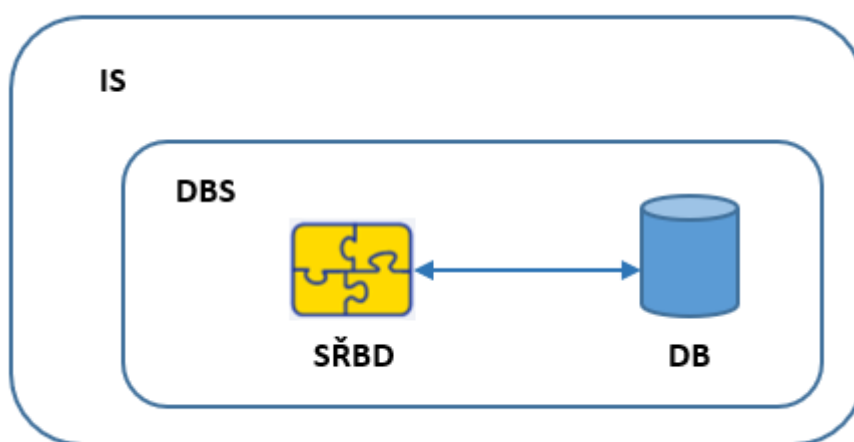
*jednotlivými prvky a objekty v databázi, schémata popisující struktury údajů a integritní omezení.“ (1)*

Jak už bylo zmíněno výše, databáze je úložiště dat, které usnadňuje jejich správu a manipulaci s daty. Data jsou uspořádána do řádků, sloupců a tabulek. A také Indexována, aby bylo usnadněno jejich nalezení. Data se aktualizují, rozšiřují a odstraňují; zatímco databázi chápeme jako soubor dat, systém pro řízení databází zajišťuje k takové databázi přístup a umožňuje s ní manipulovat.

### 3.1.4 Databázový systém

Tento pojem je často nesprávně zaměňován za SŘBD. Databázový systém můžeme slovně definovat jako data, která jsou uložena a organizována v databázi, která je řízena systémem řízení báze dat, nebo ho lze vyjádřit pomocí rovnice  $DBS = DB + SŘBD$ . Uživatel pak pro správu relační databáze používá softwarový nástroj.

Obrázek 1 - Databázový systém



*Zdroj: Vlastní zpracování*

## 3.2 Databázové modely

Databázové systémy mohou být založeny na různých databázových modelech. Model databáze určuje logickou strukturu databáze a stanovuje, jakým způsobem mohou být data uložena, organizována a manipulována. Z hlediska manipulace data a vazeb mezi nimi se zprvu začaly vyvíjet tři základní typy modelů:

- hierarchický model
- síťový model
- relační model

Později, v 90. letech, se objevil nový objektový model. Nenaplnil však očekávání a rychle přešel do modelu objektově-relačního. (3) (4) (5)

### 3.2.1 Hierarchický model

Jde o první uznávaný databázový model vytvořený společností IBM v 60. letech minulého století. Avšak dnes se jen zřídka kdy vyskytuje, vzhledem k určité provozní neefektivnosti.

Tento databázový model organizuje data do stromové struktury s jediným kořenem, ke kterému jsou připojena ostatní data. Hierarchie začíná od kořenových dat a rozšiřuje se jako strom a přidává podřízené uzly k nadřazeným uzlům.

Stromová struktura dovoluje pouze kardinalitu 1: N. (Kardinalita je podrobněji vysvětlena v kapitole 3.3.6) Hierarchický model je vhodný pro popis mnoha reálných vztahů. (3) (4) (5)

### 3.2.2 Síťový model

Síťový model je založen na hierarchickém modelu. Na rozdíl od něj dovoluje mnoho vzájemných vztahů mezi propojenými záznamy, což zahrnuje více nadřazených uzlů. Data jsou spíše uspořádána jako graf, a to především proto, že síťový model vychází z teorie matematických množin. Jeho nevýhodou je obtížná změna struktury a nepružnost. (3) (4) (5)

### 3.2.3 Relační model

Nejběžnějším modelem je relační model, který používá formát založený na relacích. Relace je reprezentována dvourozměrnou tabulkou. Každá tabulka se skládá ze sloupců a řádků, v kterých jsou tříděna data. Každý sloupec představuje atribut daného subjektu.

V řádcích, nazývá také datové n-tice, se nacházejí příslušná data konkrétní instance příslušného objektu. (3)

Souhrnem všech tabulek je potom relační databáze. Pro vedení relační SŘBD bylo definováno E. F. Coddem 12 pravidel.

1. **„Informace:** *Všechny informace v relační databázi (včetně názvů tabulek a sloupců) jsou reprezentovány explicitně jako hodnoty v tabulkovém formátu.*
2. **Zaručený přístup:** *U každé hodnoty v relační databázi je zaručeno, že bude přístupná prostřednictvím kombinace názvu tabulky, hodnoty primárního klíče a názvu sloupce.*
3. **Systematická podpora nulitních hodnoty:** *Databázový systém poskytuje systematickou podporu pro práci s nulitními hodnotami (neznámá či nepoužitelná data), které jsou odlišné od výchozích hodnot a nezávislé na jakékoli doméně.*
4. **Aktivní relační katalog dostupný online:** *Popis databáze a jejího obsahu je reprezentován na logické úrovni v tabulkové formě, a proto lze nad ním spouštět dotazy pomocí databázového jazyka.*
5. **Ucelený datový podjazyk:** *Alespoň jeden podporovaný jazyk musí mít dobře definovanou syntaxi a musí být ucelený. Musí podporovat definici dat, manipulaci s daty, integritní pravidla, autorizaci a transakce.*
6. **Aktualizování pohledů:** *Veškeré pohledy, které lze teoreticky aktualizovat, mohou být aktualizovány prostřednictvím systému.*
7. **Vkládání, aktualizace a mazání na úrovni množin:** *Databázový systém podporuje nejen získávání data na úrovni množin, ale také vkládání, aktualizace a mazání na úrovni množin.*
8. **Fyzická datová nezávislost:** *Změna fyzických přístupových metod nebo úložných struktur nemá z logického hlediska na aplikační a ad hoc programy žádný vliv.*
9. **Logická datová nezávislost:** *Změna tabulkových struktur nemá z logického hlediska na aplikační a ad hoc programy v maximální možné míře vliv.*
10. **Integritní nezávislost:** *Databázový jazyk musí být schopen definovat integritní pravidla. Tato pravidla musejí být uložena v katalogu dostupném online a nesmí existovat možnost pro jejich obejití.*

**11. Nezávislost distribuce:** První distribuce nebo redistribuce dat nemá na aplikační program a ad hoc požadavky z logického hlediska žádný vliv.

**12. Nerušitelnost:** Nesmí existovat možnost umožňující obejít integritních pravidel definovaných v databázovém jazyk pomocí jazyků nižší úrovně.“ (4)

### 3.3 Další důležité pojmy

Pro porozumění a aplikování teoretických poznatků v praktické části je třeba uvést a definovat základní pojmy týkající se databázových systémů, tvorby tabulek a jejich možnému propojení. K nejdůležitějším pojmům patří primární a cizí klíče, které jsou důležité pro udržení struktury relační databáze. Úzce na tyto klíče navazují vztahy mezi jednotlivými tabulkami.

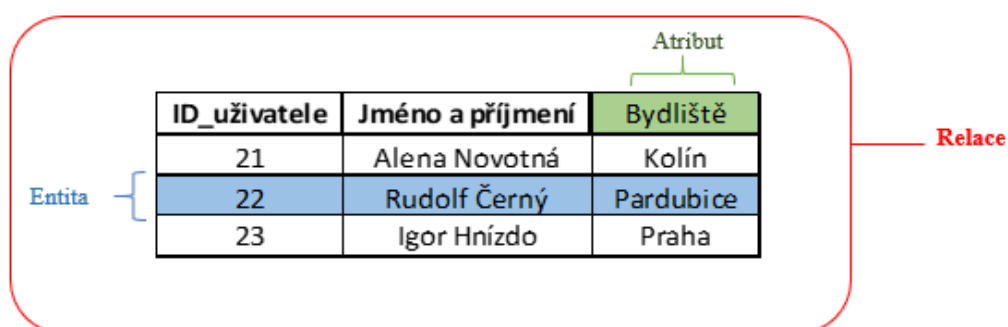
#### 3.3.1 Entita

Každá tabulka představuje entitu. Tím je myšlen jakýkoliv jedinečný, identifikovatelný a samostatný objekt.

#### 3.3.2 Atribut

Atribut vymezuje a definuje charakteristiku nebo kvantitu. V SŘBD se atribut vztahuje k databázové komponentě jako je tabulka. Atributy popisují instance v řádku databáze. Jde o nenulovou buňku v tabulce nebo spojení řádku a sloupce. Ukládá pouze jednu část dat o objektu reprezentovaném tabulkou, do které daný atribut patří. Kromě vymezeného datového typu může mít i doménu, to je množina přípustných hodnot atributu.

Obrázek 2 - Entita, atribut, relace



Zdroj: Vlastní zpracování

### 3.3.3 Klíče Relace

Klíčem v databázích je myšleno pole v tabulce, které slouží k načtení a řazení řádků na základě určitých požadavků. Klíče jsou definovány tak, aby zrychlily přístup k datům, a v mnoha případech také slouží k vytvoření vazby mezi různými tabulkami.

Relační databáze zahrnují primární i cizí klíče. Zatímco primární klíče vyžadují integritu entit a mají jedinečné hodnoty, cizí klíče zajišťují referenční integritu a vytváření spojení mezi dvěma a více tabulkami.

### 3.3.4 Primární klíč

Primární klíč je speciální sloupec tabulky relační databáze určený pro jednoznačnou identifikaci všech záznamů v tabulce. Jeho hlavní rys je jedinečná nenulová hodnota pro každý řádek dat. Primární klíč může být buď existující sloupec tabulky, nebo sloupec, který je specificky generován databází podle definované sekvence. Koncept primárního klíče je rozhodující pro efektivní relační databázi. Bez primárních klíčů by relační databáze nefungovaly.

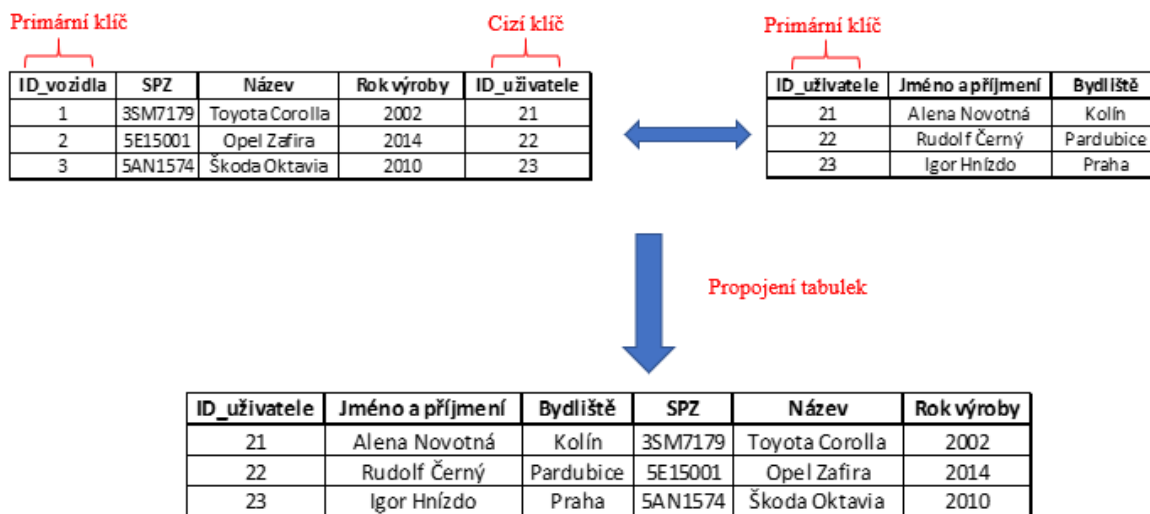
### 3.3.5 Cizí klíč

Cizí klíč je sloupec nebo kombinace sloupců v tabulce relační databáze, která poskytuje propojení mezi daty ve dvou tabulkách. Funguje jako křížový odkaz mezi tabulkami, který se odkazuje na primární klíč jiné tabulky, čímž vytvoří spojení mezi nimi.

Většina tabulek v relačním databázovém systému se řídí koncepcí cizích klíčů. Ve složitých databázích a datových skladech dochází k propojení i s více tabulkami.

Pro každý sloupec, který funguje jako cizí klíč, by měla v tabulce existovat i odpovídající hodnota. Zvláštní pozornost je třeba věnovat při vkládání a odstraňování dat ze sloupce cizího klíče. Nedbalé odstranění nebo vkládání může zničit referenční integritní omezení, tedy vztah na základě primárního a cizího klíče mezi dvěma či více tabulkami.

Obrázek 3 - Primární a cizí klíč



Zdroj: Vlastní zdroj

### 3.3.6 Kardinalita

V kontextu databází se kardinalita týká jedinečnosti hodnot dat obsažených ve sloupci relační tabulky. Vysoká kardinalita detekuje sloupec s vysokým procentem jedinečných hodnot. Kardinalita se také týká vztahů mezi tabulkami. Může být 1:1, 1: N nebo M: N.

- Vysoká kardinalita – sloupce, které mají velmi unikátní nebo neobvyklé hodnoty dat. Tabulka v databázi, kde budou uložena rodná čísla, bude mít s velkou pravděpodobností velmi vysokou kardinalitu. Každá položka dat v takové tabulce by měla být zcela jedinečná.
- Normální kardinalita – sloupce v tabulkách, které mají o něco nižší procento unikátních hodnot dat. Pokud tabulka obsahuje informace o zaměstnancích, sloupec „příjmení“ nemusí být ve všech případech unikátní.
- Nízká kardinalita – sloupce, které mají velmi málo jedinečných hodnot. V tabulce zákazníků je sloupec „pohlaví“. Tento sloupec bude mít zcela jistě nízkou kardinalitu, protože nabývá pouze hodnoty „M“ a „Ž“. (6)

Kardinalita vztahu mezi tabulkami:

- 1 : 1 – vztah mezi dvěma tabulkami, kde obě tabulky jsou spojeny pouze na jednom odpovídajícím řádku.
- 1 : N – vztah mezi dvěma tabulkami, kde řádek z jedné tabulky má více odpovídajících řádků v jiné tabulce.

- M : N – vztah mezi dvěma tabulkami, každý řádek v jedné tabulce může odkazovat na mnoho řádků v tabulce druhé a naopak. (7)

### 3.3.7 Pohled

Pohled (View) je podmnožina tabulek z databáze. Zobrazení pohledu je definován více či méně komplikovaným dotazem, který je neměnný. Údaje obsažené v pohledu jsou dynamické v závislosti na čase. Přestože pohled neukládá, data vytváří pomyslnou virtuální tabulku.

Pohled kombinuje nebo zjednodušuje data z jedné nebo více tabulek. Zabírají jen velmi málo úložného prostoru, protože databáze obsahuje pouze definici příkazu, nikoliv data. Dále mohou poskytovat výsledky pro různé výpočty (jako součet, průměr atd.). Pohledem můžeme také omezit, jaká data se budou zobrazovat uživatelům.

### 3.3.8 Formulář

V databázovém kontextu je formulář okno nebo obrazovka, který obsahuje četná pole pro zadávání dat. Každé pole obsahuje štítek s popisem tak, aby každý uživatel věděl, co pole či formulář zobrazuje. Formulář je uživatelsky přívětivější než tvorba příkazů pomocí dotazovacího jazyka pro vytváření tabulek a vkládání dat do tabulek.

## 3.4 Databázový jazyk SQL

Vytváření a úprava SQL dotazů vyžaduje základní znalost SQL. Proto je v této části popsán jazyk SQL, jeho standardní příkazy, s cílem získat obecné představy o tomto jazyku a dále tyto získané znalosti uplatnit v praktické části diplomové práce.

SQL je jazyk strukturovaných dotazů. Uživatelé ho využívají pro komunikaci s databázovým serverem pro ukládání, manipulaci a načítání dat uložených v relační databázi. Systémy řízení báze dat (SŘBD) jsou například Oracle, Microsoft Access, MySQL, IBM DB2 nebo PostgreSQL. Data se uvnitř databáze nacházejí ve formě tabulky. Tyto systémy také vyvíjejí a používají svůj vlastní dialekt SQL, který rozšiřuje funkčnost založenou na standardech SQL definovaným americkým národním standardem (ANSI). Víceméně jsou si z velké části podobné. Nadstavby jazyka SQL jsou: PL/SQL pro databáze Oracle, Transact-SQL pro Microsoft databáze nebo PL/pgSQL pro PostgreSQL databáze.

SQL hraje důležitou roli v umožnit uživatelům spravovat data uložená v SŘBD. Uživatelé mohou definovat data v databázi, manipulovat s daty, vytvářet a mazat tabulky v databázi, nastavovat oprávnění, vytvářet pohledy, procedury nebo funkce. (8)

Jazyk SQL (Structured Query Language) je založen na jazyku SEQUEL. To byl jazyk vytvořený firmou IBM k manipulaci a definici dat pro System R. Šlo o první databázový systém založený na Coddově relační databázi. První komerční databázový systém využívající SQL vyvinula společnost Oracle, dříve pod názvem Relation Software, roku 1979. Zanedlouho se SQL stal výbavou a standardní jazykem většiny relačních databázových systémů. I nadále je SQL je jedním z nejvíce používaných jazyků specifických pro práci s relační databází, která je, stejně jako SEQUEL, založena na relačním modelu E. F. Codd. (9) (10)

### 3.4.1 Struktura jazyk SQL

Pochopení struktury jazyka SQL je stěžejním prvkem této diplomové práce. SQL se skládá z několika jazykových komponentů, ty zahrnují: klauzule, výrazy, predikáty, příkazy, dotazy a mezery. Komponenty jsou znázorněny na obrázku 4.



Zdroj: Vlastní zpracování

- Příkaz na obrázku 4, vyhledává *vlastníka* auta, z tabulky *auta*, kde *staří* auta se rovná roku 2010 a *značka* auta je „Škoda“.
- Klauzule – součást příkazů a dotazů, některé jsou volitelné (např. WHERE).
- Výrazy – vyhodnotí a vrátí buď skalární hodnoty, nebo tabulky.
- Predikáty – specifické podmínky pro omezení příkazů a dotazů (AND, IN, OR).



- Příkaz – ovlivňuje zobrazení dat, relace, tok programu. Důležitým terminátorem příkazu je středník, kterým se ukončuje každý příkaz. Ačkoliv se na každé platformě nevyžaduje, je definován jako standardní část SQL gramatiky.
- Dotazy – patří k nejdůležitějšímu prvku SQL. Na základě dotazů získáváme data podle konkrétních kritérií.
- Mezery – nemají sémantiku v příkazech a dotazech SQL. Používají se k formátování SQL pro lepší čitelnost.

Standardní SQL příkazy pro komunikaci s relačními databázemi jsou CREATE, INSERT, UPDATE, ALTER, DROP, DELETE, SELECT, GRANT a další. V závislosti na jejich funkčnosti je lze rozdělit do šesti kategorií:

- Definice dat (Data Definition Language neboli DDL)
- Manipulace s daty (Data Manipulation Language neboli DML)
- Příkazy pro kontrolu transakcí (Transaction Control Statements)
- Příkazy pro řízení relace (Session Control Statements)
- Příkazy pro kontrolu systému (System Control Statement)
- Vestavěné SQL příkazy (Embedded SQL Statements) (11)

V další části bude definována podle funkčnosti příkazů pouze DDL, DML a transakční příkazy. Jedná se o nejčastěji používané příkazy, kterým je třeba porozumět.

### 3.4.2 DDL

Jazyk pro definici dat, ve zkratce DDL (Data Definition Language), umožňuje určit strukturu a vlastnosti databází. Podporuje následující akce:

- Vytváří, upravuje a maže schémata objektu
- Uděluje nebo odebírání role a privilegia
- Analyzuje informace o tabulce, indexu nebo clusteru
- Umožňuje vytvoření auditu
- Umožňuje přidání poznámek do datového slovníku

Příkazy CREATE, ALTER a DROP definují nebo upravují objekty v databázi. Příkaz CREATE je určen k definování nové tabulky, příkaz ALTER upravuje tabulky a příkaz DROP odstraňuje tabulky z databáze.

U uvedeného příkladu je tabulka vytvořena za pomoci příkazu CREATE, kde je uvedena specifikace tabulky včetně sloupců, omezení jedinečnosti a datových typů. DDL podporuje i příkazy pro jiné účely jako je vytváření a deklarování nových oprávnění pro použití databáze pomocí příkazu GRANT.

Syntaxe příkazu CREATE TABLE:

```
CREATE TABLE název_tabulky (  
    sloupec_1 DATOVÝ_TYP  
    sloupec_2 DATOVÝ_TYP  
    sloupec_3 DATOVÝ_TYP);
```

### 3.4.3 DML

Jazyk pro zpracovávání dat neboli DML (Data Manipulation Language) podporuje příkazy pro úpravu stavu databáze nebo dodatečné změny či aktualizace dat v tabulkách. Podporuje tyto příkazy: DELETE, INSERT, SELECT, UPDATE, LOCK TABLE, EXPLAIN PLAN, CALL, MERGE. DELETE, INSERT, SELECT a UPDATE, které umožňují vývojářům manipulovat s uloženými daty v databázi.

INSERT vkládá nové řádky do tabulky. SELECT vybírá data, která jsou uložena v tabulce. Update slouží k úpravě a aktualizaci konkrétních záznamů, ať už se jedná o datový typ, integritní omezení nebo změnu názvu. DELETE, odstraňuje existující záznamy v tabulce.

Ostatní příkazy LOCK TABLE, EXPLAIN PLAN, MERGE a CALL, poskytují vývojářům větší kontrolu nad tabulkami. Mohou do nich lépe nahlížet, optimalizovat nebo slučovat výrazy. Příkaz LOCK TABLE automaticky uzamyká mechanismus v SQL, aby zabránil přístupu k datům nebo jejich manipulaci ostatními uživateli. Příkaz Call volá určité funkce nebo uložené procedury v databázi. (12) (13)

### 3.4.4 Transakční příkazy

Jazyk SQL poskytuje sadu příkazů, které vývojářům dávají kontrolu nad databázovými transakcemi – COMMIT, ROLLBACK, SAVE, SET TRANSACTION. Příkaz COMMIT slouží k uložení změn vyvolaných transakcí do databáze. Uloží všechny transakce od posledního příkazu COMMIT nebo ROLLBACK. V případě selhání je použit příkaz

ROLLBACK, který navrátí databázi do předchozího stavu. SAVE ukládá aktuální stav databáze, po použití ROLLBACK nebo příkazu SET TRANSACTION. Ten umožňuje vývojáři podrobněji zadat vlastnosti transakce. Zajišťuje, v jakém bodě má být obnovena databáze v případě selhání transakce. (14) (15)

### 3.5 Procedurální jazyk PL/SQL

Jazyk SQL byl rozšířen o procedurální jazyk. PL/SQL pro databáze Oracle, Transact-SQL pro Microsoft databáze nebo PL/pgSQL pro PostgreSQL databáze. Každé z těchto rozšíření umožňuje vývojářům kombinovat příkazy SQL s procedurálními jazykovými konstrukty, jako jsou funkce, procedury nebo Triggery.

Funkce, procedury a různé definice výjimek mohou být seskupeny do tzv. balíků (package). Výhodou jejich použití je souhrn všech potřebných komponentů, pro řešení problému, do jednoho celku. To usnadní práci hlavně při přenosu komponentů do jiné databáze.

Tyto metody mohou být definovány za pomoci příkazu CREATE (DDL), volány pomocí CALL (DML) nebo příkazem PL/SQL kódu. Mnoho SŘBD podporuje toto procesní rozšíření, ale každá z nich se liší v podporovaných příkazech. SQLite nepodporuje funkce a uložené procedury, zatímco PL/SQL pro databázi Oracle ano.

#### 3.5.1 Struktura příkazu PL/SQL

Základní jednotky PL/SQL (funkce, procedury), které tvoří program, jsou tzv. logické bloky, které mohou obsahovat libovolný počet vnořených dílčích bloků. Každý blok se skládá ze tří dílčích částí. První část začíná prvkem DECLARE. Jedná se o volitelnou část, kde se definují všechny proměnné, kurzory, podprogramy a další prvky, které mají být v bloku používány. Druhá část je uzavřena klíčovými slovy BEGIN a END. Tato část je povinná. Zde se nacházejí příkazy PL/SQL kódu. Tato sekce by měla mít alespoň jeden spustitelný řádek. Třetí a poslední část pracuje s výjimkami. Začíná slovem EXCEPTION, je volitelná a zpracovává chyby, které mohou nastat v bloku. (16) (17)

Syntaxe vytvoření procedury/funkce:

**DECLARE**

-- Zde se deklarují proměnné.

**BEGIN**

-- Zde leží tělo bloku, v kterém běží procedury/funkce a SQL kód. Tato sekce je jako jediná povinná.

**EXCEPTION**

-- Oblast zpracovávání výjimek.

**END;**

### 3.5.2 Funkce a procedury

Funkce a procedury jsou si velmi podobné. Hlavní rozdíl mezi nimi je v návratové hodnotě. Funkce vykoná danou logiku a na rozdíl od procedury vrací hodnotu předem definovaného typu. Jak funkce, tak procedury můžeme zobecnit přidáním vstupních parametrů. Tím pádem mohou provádět stále stejný kód, ale pokaždé s jinými daty.

Definice příkazu procedury:

**CREATE PROCEDURE** název\_procedury (parametr IN datový\_typ, ...) AS

**BEGIN**

-- vykonávány kód

**END;**

Definice příkazu funkce:

**CREATE FUNCTION** název\_funkce (parametr IN datový\_typ, ...) AS

**RETURN** datový\_typ\_návratové\_hodnoty

**BEGIN**

-- vykonávány kód

**RETURN** hodnota\_k\_navrácení;

**END;**

### 3.5.3 Trigger

Databázový Trigger (z angličtiny – spínač, spouštěč) je speciální uložená procedura, která se automaticky spouští nad tabulkou, pokud v ní dojde k určité akci. Triggery jsou definovány, jak při změnách údajů v tabulce, tzn. při volání DML (INSERT, UPDATE, DELETE) nebo DDL operace (ALTER, DROP, CREATE), tak při operacích v databázi. Spuštění a zastavení databáze, přihlášení a odhlášení uživatele do databáze nebo vznik chyby v databázi.

Triggery pomáhají správci databáze zajistit kontrolu, zda byla dokončena určitá akce nebo kdo provedl poslední úpravu dat. Na rozdíl od procedur a funkcí má jistá omezení, jako například délku příkazu, a nelze v nich používat příkazy (COMMIT, ROLLBACK). Z toho

důvodu je vhodné Triggery používat u méně složitých příkazů a na ostatní úkony používat funkci či proceduru.

Příklad použití Triggeru:

**Obrázek 5 - Příkaz vytvoření Triggeru**

```
CREATE OR REPLACE TRIGGER trg_prihlaseni
AFTER LOGON ON DATABASE
BEGIN
    INSERT INTO kontrola_prihlaseni (datum, uzivatel) VALUES (SYSDATE, USER);
END;
```

Vytvoř nebo nahraď trigger s názvem „trg\_prihlaseni“ po přihlášení do databáze.

Mezi „BEGIN“ a „END“ je samotný kód spouště.

Vlož záznamy „SYSDATE“ (aktuální datum) a USER (jméno uživatele) do tabulky s názvem „kontrola\_prihlaseni“.

*Zdroj: Vlastní zpracování*

### 3.5.4 Sekvence

Sekvence je databázový objekt, který generuje celočíselnou sekvenci, je používán pro vyplnění sloupců čísel primárního klíče.

Pro vytvoření sekvence se použije příkazu DDL, jeho syntaxi můžeme vidět na obrázku 5.

Příkaz začíná formulací CREATE SEQUENCE, následuje název sekvence. Pro lepší přehled je sekvence nejčastěji pojmenována podle tabulky, ke které se váže. Sekvence má několik atributů:

- **START WITH** – za atribut se zadává celočíselná hodnota, jedná se o číslo, které bude prvním generovaným číslem sekvence,
- **INCREMENT BY** – tento atribut požaduje nenulovou číselnou hodnotu. Zadané číslo slouží jako interval, o který se bude sekvence zvyšovat či snižovat. Pokud je klauzule vynechána, je výchozí interval 1,
- **MAXVALUE/NOMAXVALUE** – pomocí těchto atributů lze nastavit maximální horní hranici pro sekvenci čísel, nebo naopak maximální horní hranice nebude uvedena,
- **MINVALUE/NOMINVALUE** – podobně jako MAXVALUE se používá i MINVALUE s tím rozdílem, že nastavuje minimální dolní hranici sekvence,
- **CACHE/NOCACHE** – hodnota atributu určuje počet celých čísel v mezi paměti, případně použití atributu NOCACHE nejsou hodnoty předem přiděleny. Pokud

není uvedena ani jedna klausule databáze ukládá do paměti cache 20 pořadových čísel,

- **CYCLE/NOCYCLE** – je-li nastaven **CYCLE**, poté sekvence cyklicky generuje hodnoty po dosažení maximální či minimální hodnoty. Jakmile vzestupná sekvence dosáhne své maximální hodnoty, začne generovat znovu od minimální hodnoty a naopak. Pokud je nastaven příznak **NOCYCLE**, nemůže dojít po dosažení minimální nebo maximální hodnot k vygenerování další hodnot. Toto je i výchozí nastavení v případě, že není atribut vyplněný.

K zajištění automatického vložení další hodnoty sekvence do tabulky při založení nového záznamu, je třeba definovat Trigger.

Příklad použití sekvence:

**Obrázek 6 - Příkaz sekvence pro Oracle databázi**

CREATE SEQUENCE název_sekvence			
START WITH	1000	}]	Počáteční hodnota, na které začne sekvence.
INCREMENT BY	1	}]	O kolik se navyšuje předchozí hodnota.
NOMAXVALUE		}]	Není definovaná žádná maximální a minimální hodnota sekvence.
NOMINVALUE		}]	
NOCACH		}]	K výpočtu hodnot není třeba vyrovnávací paměti.
NOCYCLE;		}]	Sekvence se není cyklická.

*Zdroj: Vlastní zpracování*

### 3.5.5 Proměnné

Z předešle kapitoly je jasné, že proměnné se definují v deklarační sekci bloku PL/SQL. Pro deklarování proměnných je třeba proměnou pojmenovat, uvést datový typ a ukončit středníkem. Lze také explicitně přidat datové omezení, které je vloženo v závorkách za datovým typem proměnné.

Příklad definování proměnných v PL/SQL bloku:

```
DECLARE  
v_jmeno VARCHAR2(20);  
v_prijmeni VARCHAR2(20);  
n_poradi NUMBER;  
d_pronajem DATE;  
BEGIN
```

```
NULL;  
END;
```

V PL/SQL je jedním z nejběžnějších úkolů výběr hodnot ze sloupců tabulky do množiny proměnných. V případě změny datových typů sloupců v tabulce je nutné upravit i program PL/SQL, aby byly datové typy proměnných kompatibilní. V takovém případě PL/SQL nabízí užitečnou funkci, a to použití tzv. ukotvení za pomoci **%TYPE**, které je vloženo za definovanou proměnnou. V tomto případě je k proměnné přiřazen stejný datový typ, který je uveden u sloupce v tabulce.

### 3.5.6 Řídicí struktury

#### 3.5.6.1 Podmínky IF a CASE

Řídicí struktury umožňují programátorovi zvolit jednu nebo více podmínek, které mají být vyhodnoceny a provedeny. Podmínkou může být cokoli, co vyhodnotí logickou hodnotu Boolean. Pod textem je uvedena struktura podmínky IF nejen pro PL/SQL. Tak neplatí jen pro jazyk PL/SQL, ale pro většinu programovacích jazyků.

```
IF podmínka_1 THEN  
  -- příkaz_1  
ELSIF podmínka_2 THEN  
  -- příkaz_2  
ELSE  
  -- příkaz_3  
END IF;
```

Příkaz CASE je velmi podobný příkazu IF. Avšak CASE na rozdíl od IF vybírá ze sekvence podmínek. Vyhodnotí jeden výraz a porovnává je s několika dalšími potenciaálními hodnotami. Pokud vyhodnotí víc Booleonovských výrazů, pak vybere první, který je pravdivý. (18) (19)

Struktura podmínky CASE:

```
CASE výraz  
  WHEN podmínka_1 THEN příkaz_1  
  WHEN podmínka_2 THEN příkaz_2  
  ELSE příkaz_3;  
END CASE;
```

#### 3.5.6.2 Příkaz LOOP

Příkaz LOOP umožňuje provést příkaz nebo skupinu příkazů několikrát za sebou. Na obrázku 20 je vidět, jak příkaz probíhá:

Typy příkazu LOOP:

- Základní příkaz LOOP – při každé iteraci je provedena sekvence příkazů, cyklus je možné přerušit příkazem EXIT (skončí provádění cyklu) a EXIT WHEN (ukončení cyklu, pokud je plněna daná podmínka).
- WHILE LOOP – opakuje příkaz nebo skupinu příkazů, pokud je daná podmínka pravdivá.
- FOR LOOP – řízené opakování, je-li potřeba povést určitý počet opakování. Pokud příkaz obsahuje klauzuli REVERSE, probíhá cyklus sestupně od největšího čísla k nejmenšímu.
- Vnořený příkaz LOOP – použití jedné smyčky uvnitř jiné. (19) (20)

Jednoduchý příklad příkazu FOR LOOP, kde příkaz dbms\_output.put\_line zobrazuje výstup smyčky:

```
DECLARE
  n_cislo NUMBER;
BEGIN
  FOR n_cislo IN 10 .. 13 LOOP
    dbms_output.put_line ('Hodnota proměnné: '|| n_cislo);
  END LOOP;
END;
```

### 3.5.7 Kurzor

Hlavním účelem jazyka PL/SQL je co nejjednodušší a nejúčinnější dotazování a změna obsahu tabulek v databázi. Jsou-li volána data z tabulky, pokaždé jsou výsledky vrácené SQL příkazem SELECT kurzory. Správu kurzoru DLM příkazů zpracovává databáze Oracle, ale PL/SQL nabízí několik způsobů, jak definovat a manipulovat kurzory s příkazem SELECT.

Rozlišujeme dva typy kurzorů: implicitní a explicitní. Jestliže je kurzor definován uvnitř bloku deklarace, jde o explicitní kurzor. Příkazy DML uvnitř libovolného bloku jsou implicitními kurzory. Implicitní kurzor je také vytvořen pokaždé, když je použit SELECT s příkazem INTO nebo SELECT uvnitř příkazu pro LOOP FOR. (20)

Příklad explicitního kurzoru:

```
DECLARE
CURSOR c IS
  SELECT      id
              jmeno
              prijmeni
FROM        uzivatele;
```



```
BEGIN  
FOR i IN c LOOP  
    dbms_output.put_line ('Uživatel '|| i.jmeno || i.primeni);  
END LOOP;  
END;
```

## 3.6 Oracle SQL Developer

Praktická část byla vypracována převážně v aplikaci Oracle SQL Developer, proto je vhodné tuto aplikaci přiblížit.

Oracle SQL Developer je integrované vývojové prostředí pro práci s SQL v databázích Oracle. Tento produkt Oracle poskytuje zdarma jako volně šiřitelnou aplikaci. Splňuje všechny požadavky a procesy spojené se správou databáze a usnadňuje její správu. Nacházejí se zde základní funkce pro manipulaci dat a funkce pro automatizaci práce s databází. Uživatelům umožňuje testování databázových objektů nebo celých databází. Pro kontrolu a ladění objektů napsaných za pomoci PL/SQL je tu debugger. (21)

## 3.7 Report

V závěru praktické části práce byl vytvořen report na základně předchozí analýzy požadavku uživatelů, kteří chtěli informace stáhnout a tisknout ve formě PDF pro archivace v papírové podobě. Pro tvorbu reportu samotného byl použit softwarový nástroj Jaspersoft Studio.

### 3.7.1 Jaspersoft Studio

Produkt Jaspersoft Studio je vytvořen pomocí jazyka Java a jedná se o rozhraní pro programování aplikací. Jde o grafický designový nástroj s otevřeným zdrojovým kódem (XML), který vývojářům umožňuje vytvářet návrhy reportů a rozvrhovat vizuální komponenty jako jsou grafy a tabulky. Také poskytuje přívětivé rozhraní pro koncové uživatele, kteří přehledy zobrazují.

## **4 Vlastní práce**

### **4.1 Vymezení problému**

Ve firmě VUSTE-APIS, s.r.o., ve které autorka pracuje, došlo k implementaci nové verze podnikového informačního systému VAMAX, který podnik sám vyvíjí a spravuje. VAMAX je systém pro řízení a administraci podnikových nebo projektových procesů. Jeho nová verze prošla úpravou jak po vizuální, tak databázové stránce. To je například upravení kódu a vytvoření nových formulářů: uživatelé, správa adresářů a souborů. Dále byly přidány nové sekce: plánování kapacit, splátkový kalendář, pronájmy a také kniha jízd, která je předmětem této diplomové práce.

Z uživatelského hlediska lze na systém VAMAX nahlížet jako na škálovatelnou stavebnici. Do portfolia VAMAXu je možné přidávat další funkční bloky, které jsou pro daný podnik důležité. Společnost VUSTE-APIS v současné době zvažuje obchodní aktivity vedoucí k prodeji systému VAMAX. Z tohoto hlediska je kniha jízd velmi důležitou položku. Potenciální klienti VUSTE-APIS si mohou vybrat, které bloky VAMAXu zakoupí a které ne. Tedy i blok knihy jízd, který je nyní integrální součástí systému VAMAX společnosti VUSTE-APIS, může být volitelnou a prodejnou položkou pro externí zákazníky této společnosti.

Firma za spolupráce autorky práce provedla analýzu stávajícího stavu a vytvořila návrh nového řešení knihy jízd v elektronické podobě. Ta se doposud zapisovala pouze do excelové tabulky. Nejprve byl vypracován logický model, poté následovala realizace tabulek a tvorba balíčků za pomoci databázového programovacího jazyka SQL a jeho nadstavby PL/SQL. Pro uživatelsky snadnější zápis dat do databáze byly navrženy formuláře prostřednictvím grafického prostředí Forms Developer v informačním systému VAMAX. V závěru je vytvořen report v softwaru Jaspersoft studio generující data ve formátu PDF v podobě pro tisk A4 pro archivaci v papírové podobě a je vytvořena uživatelská příručka k příslušným formulářům.

### **4.2 Framework**

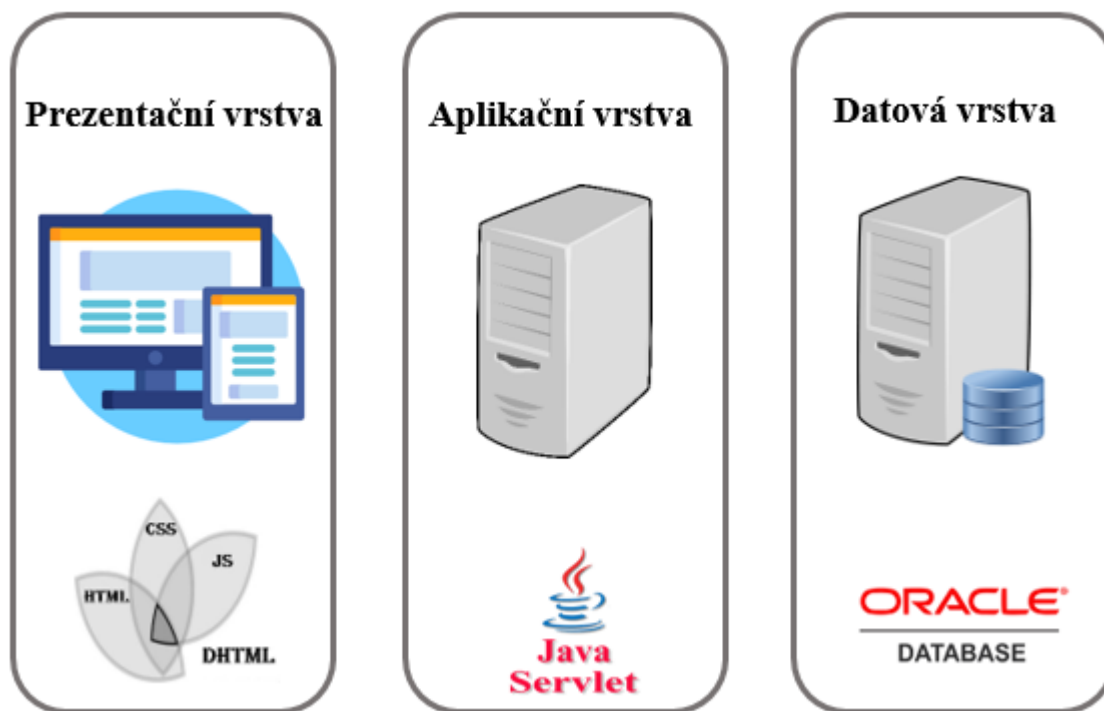
Třívrstvý firemním framework vychází z architektury Oracle Forms a slouží pro tvorbu webových aplikací pevně spojených s databází Oracle. Poskytuje nástroje pro rychlé vyvíjení komplexních formulářů, zobrazení dat a dalších komponentů v prostředí prohlížeče. Hlavním cílem frameworku je poskytnout snadnější použití komponentů, které jsou

kombinovány a deklarovány způsobem pro rychlý vývoj aplikací. Oproti klasické business logice, která je vedena jako samostatná vrstva, zde zůstávají jak data, tak business logika v databázi. Aplikační vrstva jen zde velice tenká. Zprostředkovává a zajišťuje komunikaci mezi prezentační vrstvou a databází. To je možné právě díky databázi Oracle, která se dá programovat, například u databáze MySQL by toto nebylo možné. Velkou výhodou této logiky je rychlost a možnost programování za běhu.

Frameworková architektura

- Datová vrstva (Oracle databáze)
- Business logická struktura (Oracle PL/SQL)
- Tenká aplikační vrstva (Java Servlet)
- Prezentační vrstva (webový prohlížeč)

Obrázek 7 - Třívrstvá frameworková architektura



Zdroj: Vlastní zpracování

### 4.3 Požadavky na nový systém

Prvním krokem je definice základních požadavků na nový systém. Požadavky autorka diskutovala s vedoucím oddělení ICT firmy VUSTE-APIS, který vytvoření knihy jízd inicioval. Obecně vzato je možné požadavky rozdělit do pěti kategorií:

- Požadavky na parametry vyžadované od uživatelů
- Požadavky na uživatelské rozhraní
- Logické požadavky pro vytvoření DB
- Požadavky na HW
- Požadavky na SW

Požadavky na parametry vyžadované od uživatelů a uživatelského rozhraní vyplývají ze stávající excelové tabulky, kterou zaměstnanci firmy VUSTE-APIS byli nuceni vyplňovat před implementací nového rozšíření systému VAMAX o knihu jízd (obrázek 8). V systému není potřeba již položky jako „podpis řidiče“ a „schválil“, neboť tyto jsou součástí schvalovacího procesu ve VAMAXu. Požadavky na uživatelské rozhraní jsou z principu definovány celkovým grafickým formátem systému VAMAX. Tedy tlačítka a formuláře mají design stejný jako ostatní komponenty systému VAMAX. Logické požadavky pro tvorbu DB jsou popsány v kapitole 4.4 Analýza a návrh tvoří stěžejní část diplomové práce. Požadavky na HW (DB) nebylo nutné řešit z těchto důvodů:

- firma VUSTE-APIS používá cloudové řešení,
- kniha jízd a registr vozidel nepředstavuje z principu omezení,
- pokud dojde k nutnosti navýšení diskového prostoru nebo paměti, firma má nastaven proces, jak zažádat o výkonnější HW prostředky.

Obrázek 8 - MS Excel s požadavky na zadávané parametry

	<b>Kniha jízd firemního vozidla</b>	
	Předmět použití : Záznam zákonem stanovených informací o jízdě firemním vozidlem Vazba na dokument : Směrnice č. 1/2006      Garant formuláře : Roman Černý      Označení : ŘS91F09	

SPZ:	<b>7A6 5590</b>
Typ vozidla:	<b>Octavia II 1.9 TDI / 77 kW</b>

Rok:	<b>2015</b>	Měsíc:	<b>leden</b>
------	-------------	--------	--------------

Jméno řidiče	<b>Ing. Vojtěch Jiříčka</b>
--------------	-----------------------------

Osobní číslo řidiče:	<b>7869</b>
----------------------	-------------

Datum	Místo počátku - cíl - místo ukončení cesty	Od	Do	Ujeto km		Účel jízdy
				Služebně	Soukromě	
1.1						
2.1						
3.1						
4.1						
5.1						
6.1	VUSTE - DATART - VUSTE	10:00	12:00	50		Pracovní
7.1						
8.1						
9.1						
10.1						
11.1						
12.1						
13.1						
14.1						
15.1						
16.1						
17.1						
18.1						
18.1						
19.1						
20.1	VUSTE-SUNNYSOFT-VUSTE	15:00	17:00	50		Pracovní
21.1						
22.1						
23.1						
24.1						
25.1						
26.1						
27.1						
28.1	VUSTE-SERVIS-VUSTE	8:00	9:00	20		Servis vozu
29.1						
30.1						
31.1						
<b>Celkem</b>				<b>120</b>	<b>500</b>	
				<b>96 042</b>	<b>620</b>	<b>96 662</b>

Poznámky: Vzor pro datum: 12.6      Voz SPZ: 3A5 2549 Vzor pro místo ....: VUSTE-APIS - SZIF - ČSOB - VUSTE-APIS Vzor pro čas: 9:30 14:40 Pro více služeb, cest v jednom dni lze použít více řádek. U soukromé cesty se vyplňuje pouze datum a ujeté km.	Podpis řidiče:	Schválil:
	Datum:	Datum:
Účel služební jízdy:	Pracovní      Jiný	Stav ujetých km na počátku roku (zařazení do provozu):      96042
	Obchodní      Servis vozu	Stav ujetých km na konci roku:

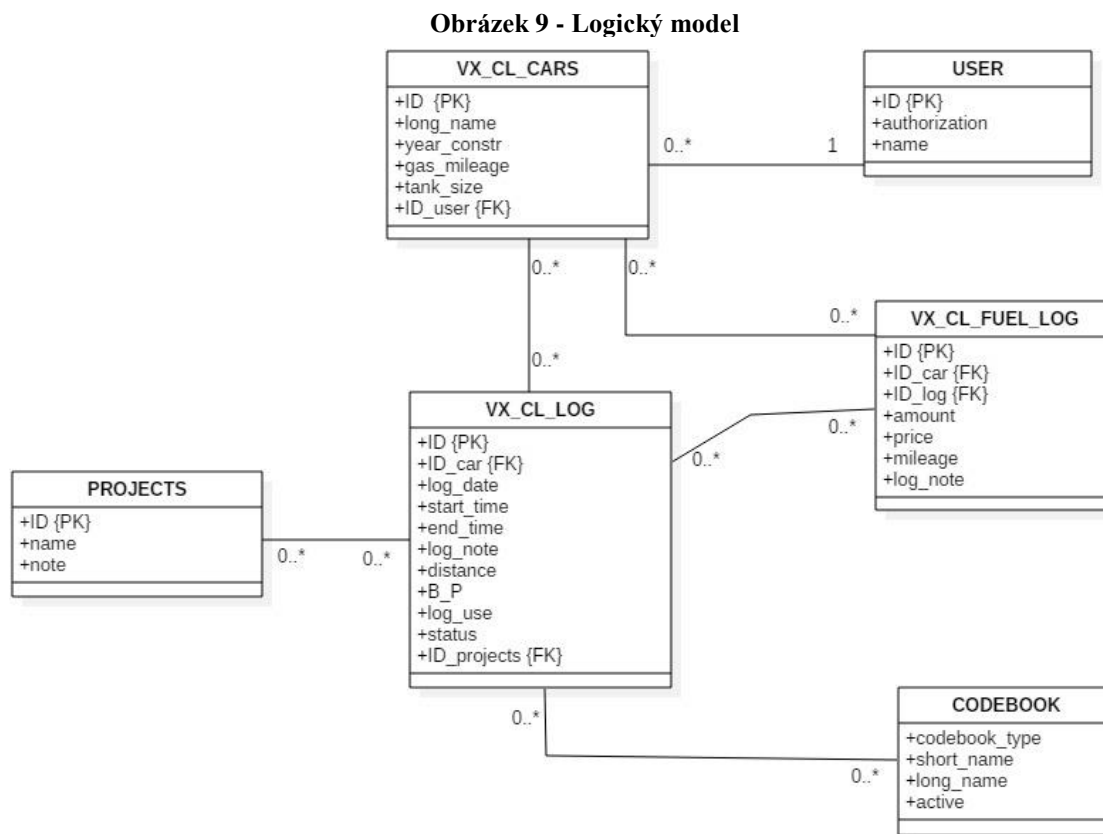
Zdroj: Vlastní zpracování

## 4.4 Analýza a návrh

Tato část popisuje základní technické aspekty knihy jízd. Hrubý návrh funkcí, logický modelu dat a strukturu dat v jednotlivých tabulkách, ze kterých vychází samotná realizace knihy jízd.

### 4.4.1 Logický návrh databáze

Návrh logického modelu je prvním bodem tvorby databáze. Zajišťuje lepší přehled dat, která jsou logicky uspořádána do skupin. Logickým modelem se snižuje či eliminuje duplicita dat. Níže uvedené schéma představuje logický návrh databáze vytvořené v softwarové aplikaci StarUML. Obsahuje propojení jednotlivých tabulek, jejich kardinalitu, primární a cizí klíče.



Zdroj: Vlastní zpracování

### 4.4.2 Datový slovník

Každý sloupec v databázové tabulce musí mít název a uvedený datový typ. Vývojář se při vytváření tabulky rozhoduje, jaký typ dat bude uložen uvnitř každého sloupce. Datový

typ je vodítkem, které umožňuje pochopit, jaký typ dat se očekává uvnitř každého sloupce, a také identifikuje, jak bude programátor dál pracovat s uloženými daty.

**Tabulka 1 - Datový slovník**

Tabulka	Název sloupce	Datový typ	Délka (byte)	Komentář	Další omezení
<b><i>VX_CL_CARS</i></b>	ID	number		PK	not null
	REG_PLATE	char	8	SPZ	not null
	LONG_NAME	vvarchar2	50	název	
	YEAR_CONST	number		rok výroby	
	GAS_MILEAGE	number		průměrná spotřeba	
	TANK_SIZE	number		velikost nádrže	
	ID_USER	number		zodpovědný pracovník	
<b><i>VX_CL_LOG</i></b>	ID	number		PK	not null
	ID_CAR	number		FK	not null
	LOG_DATE	date		datum	
	START_TIME	date		čas startu	
	END_TIME	date		čas cíle	
	LOG_NOTE	vvarchar2	70	popis	
	DISTANCE	number		vzdálenost	
	B_P	char	2	typ jízdy – B služební, P soukromé	
	LOG_USE	vvarchar2	50	účel jízdy – výběr z číselníku (CODEBOOK)	
	ID_PROJECTS	number		Projekt	
	STATUS	char	2	RO-rozpracováno, SM-schváleno	
<b><i>VX_CL_FUEL_LOG</i></b>	ID	number		PK	not null
	ID_CAR	number		FK	not null
	ID_LOG	number		FK	not null
	AMOUNT	number		množství paliva	
	PRICE	number	10	cena	
	MILEAGE	number		stav tachometru	
	LOG_NOTE	vvarchar2	70	poznámka (čerpací stanice)	
<b><i>USER</i></b>	ID	number		PK	not null
	NAME	vvarchar2	20	Jméno + příjmení	not null
	LOGIN	vvarchar2	30	login	not null
	STATE	char	2	role	
	BAD_LOGINS	number		počet špatných přihlášení	
	MUST_CHANGE_PASS	char	1	změna hesla A/N A – ano, N – ne	
	TEL	vvarchar2	13	telefonní číslo	not null
	MAIL	vvarchar2	100	e-mail	not null
<b><i>PROJECTS</i></b>	ID	number		PK	not null

	SHORT_NAME	varchar2	20	zkratka	not null
	LONG_NAME	varchar2	80	název	not null
	DAT_OD	date		datum od	not null
	DAT_DO	date		datum do	not null
<b>CODEBOOK</b>	CODEBOOK_TYPE	varchar2	10	řazení číselníků	not null
	SHORT_NAME	varchar2	60	zkratka	not null
	LONG_NAME	varchar2	60	název	
	ACTIVE	number	1	1 - aktivní; 0 - neaktivní	

Zdroj: Vlastní zpracování

#### 4.4.3 Návrh funkcí formulářů

Jedná se o hrubý návrh funkcí formulářů, který je jedním z klíčových bodů pro možnost implementace samotné knihy jízd. Vychází z analýzy požadavků uživatelů, správné podoby z hlediska právního předpisu bezpečnosti práce a náležitostí k prokázání možnosti uplatnění nároku na odpočet DPH.

Evidence vozů

- F\_P\_VX\_CL\_CARS
  - Přehled o všech firemních vozidlech, editace možná po kliknutí na tlačítko Editace (otevření formuláře F\_E\_VX\_CL\_CARS)
  - Oba formuláře budou obsahovat SPZ, název vozu, rok výroby, průměrnou spotřebu, zodpovědného pracovníka, velikost nádrže

Knihy jízd

- F\_P\_VX\_CL\_LOG
  - Formulář obsahuje přehled s filtrovacími poli: vůz, měsíc (výběr 1-12), rok (předvyplněný aktuální rok)
  - Tlačítka Nový a Editace budou otvírat editační formulář (dostává parametr ID vybraného vozu)
  - Formulář bude zobrazovat sloupce – datum, čas startu a cíle (jeden sloupec oddělen pomlčkou), popis jízdy, ujetou vzdálenost, typ jízdy a účel jízdy
  - Po stisknutí tlačítka Report se vytiskne report knihy jízd pro daný měsíc
- F\_E\_VX\_CL\_LOG
  - Bude zobrazovat SPZ vozu a oprávněnou osobu (nebude editační)



- Typ jízdy a účel jízdy budou vybírány ze seznamu předem daných hodnot
- Formulář dále obsahuje tabulku VX\_CL\_FUEL\_LOG (filtrovanou podle ID\_LOG), zde bude možné mazat a přidávat záznamy, pokud není záznam ve stavu schválený, tzn. pouze pro čtení

## 4.5 Implementace

Po určení datové struktury, logického modelu a jednotlivých funkcí formulářů, realizovala autorka návrh knihy jízd. Implementaci si rozdělila do pěti hlavních bodů:

- 1) Vytvoření třech tabulek (ke každé tabulce se váže trigger a sekvence), třech pohledů, přidání záznamů do tabulky CODEBOOK,
- 2) tvorba přehledových a editačních formulářů po vizuální stránce, propojení a načítání dat z databázi do formulářů,
- 3) programování package v databázi vázaných k formulářům,
- 4) tvorba reportu – propojení s databází a následné definování vzhledu,
- 5) sepsání uživatelské příručky k jednotlivým formulářům.

### 4.5.1 Tvorba tabulek

Ukázka kódu pro vytvoření tabulky VX\_CL\_LOG s příslušnými komentáři k jednotlivým sloupcům, sekvencí a triggeru:

```
CREATE TABLE vx_cl_log
( ID NUMBER NOT NULL ENABLE,
 ID_car NUMBER,
 log_date DATE,
 start_time DATE,
 end_time DATE,
 log_note VARCHAR2(70 BYTE),
 distance NUMBER,
 b_p CHAR (2 BYTE),
 log_use VARCHAR2(50 BYTE),
 ID_projects NUMBER,
 status CHAR (2 BYTE) DEFAULT 'RO',
 ID_documents NUMBER,
 CONSTRAINT vx_cl_log_PK PRIMARY KEY (ID)
 CONSTRAINT VX_CL_CARS_FK FOREIGN KEY (ID_car)
 REFERENCES USER (ID)
 CONSTRAINT projects_FK FOREIGN KEY (ID_projects)
```



```

CREATE OR REPLACE FORCE VIEW qvx_cl_log_overview AS
SELECT
  a.rowid AS ax____rowid,
  a.id,
  a.id_car,
  a.distance,
  a.log_note,
  a.log_date,
  a.start_time || ' - ' || a.end_time AS a_time,
  DECODE(B_P, 'B', 'Služební', 'P','Pracovní') AS b_p_txt,
  DECODE(LOG_USE, 'PRAC', 'Pracovní', 'JINY','Jiný', 'OBCHOD', 'Obchodní', 'SERVIS', 'Servis vozu')
AS log_use_txt,
  pax_users.getinfo (c.id_user) AS user_name,
  c.reg_plate,
  c.long_name,
  SUM(b.price) AS S_price,
  SUM(b.amout) AS s_amout
FROM vx_cl_log a
LEFT OUTER JOIN vx_cl_fuel_log b ON a.id = b.id_log
INNER JOIN vx_cl_cars c ON a.id_car=c.id
GROUP BY a.rowid, a.id, a.id_car, a.distance, a.log_note, c.id_user, c.reg_plate, c.long_name,
a.start_time, a.end_time, a.log_date, a.b_p, a.log_use
ORDER BY a.log_date DESC;

```

Obrázek 11 - Pohled QVX\_CL\_LOG\_OVERVIEW

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID
1	AX____ROWID	ROWID ()	Yes	(null)	1
2	ID	NUMBER	No	(null)	2
3	ID_CAR	NUMBER	Yes	(null)	3
4	DISTANCE	NUMBER	Yes	(null)	4
5	LOG_NOTE	VARCHAR2 (70)	Yes	(null)	5
6	LOG_DATE	DATE	Yes	(null)	6
7	A_TIME	VARCHAR2 (13)	Yes	(null)	7
8	B_P_TXT	VARCHAR2 (10)	Yes	(null)	8
9	LOG_USE_TXT	VARCHAR2 (11)	Yes	(null)	9
10	USER_NAME	VARCHAR2 (4000)	Yes	(null)	10
11	REG_PLATE	VARCHAR2 (10)	No	(null)	11
12	LONG_NAME	VARCHAR2 (50)	Yes	(null)	12
13	S_PRICE	NUMBER	Yes	(null)	13
14	S_AMOUT	NUMBER	Yes	(null)	14

Zdroj: Vlastní zpracování

Do již vytvořené tabulky CODEBOOK v databázi byla přidána data, z kterých uživatel bude vybírat do řádku LOG\_USE tabulky VX\_CL\_LOG. Ukázka kódu insertovaných dat:

```
INSERT INTO codebook (codebook_type, short_name, long_name, aktive)
VALUES ('CL_LOG_USE', 'PRAC', 'PRACOVNÍ', 1);
INSERT INTO codebook (codebook_type, short_name, long_name, aktive)
VALUES ('CL_LOG_USE', 'JINY', 'JINÝ', 1);
INSERT INTO codebook (codebook_type, short_name, long_name, aktive)
VALUES ('CL_LOG_USE', 'OBCHOD', 'OBCHODNÍ', 1);
INSERT INTO codebook (codebook_type, short_name, long_name, aktive)
VALUES ('CL_LOG_USE', 'SERVIS', 'SERVIS VOZU', 1);
```

Obrázek 12 - Tabulka CODEBOOK

	CODEBOOK_TYPE	SHORT_NAME	LONG_NAME	ACTIVE
1	CL_LOG_USE	JINY	Jiný	1
2	CL_LOG_USE	OBCHOD	Obchodní	1
3	CL_LOG_USE	PRAC	Pracovní	1
4	CL_LOG_USE	SERVIS	Servis vozu	1

Zdroj: Vlastní zpracování

#### 4.5.2 Vizuální podoba formulářů

Po tvorbě databázových tabulek autorka vytvořila grafickou podobu formulářů za pomoci tzv. Forms developeru. Jedná se o komponentu pro editaci XML, která definuje formuláře prostřednictvím grafického prostředí. Po vytvoření nového formuláře je založena pouze základní šablona s pracovní plochou. Automaticky se vytváří package v databázi se stejným názvem jako formulář, kde se pomocí procedur a funkcí formulář ovládá. Pro lepší přehled značí písmeno „P“, v názvu formuláře, přehledový formulář a písmeno „E“ editační jako editační formulář.

##### Přehledové formuláře

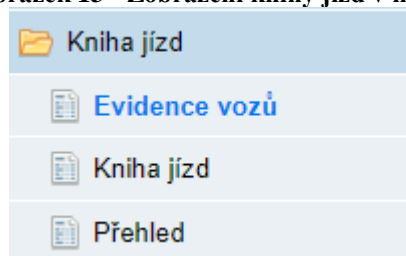
- Evidence vozů – F\_P\_VX\_CL\_CARS
- Kniha jízd – F\_P\_VX\_CL\_LOG
- Přehled vozidel F\_P\_VX\_CL\_OVERVIEW (jen pro prohlížení)

##### Editační formuláře

- Evidence vozů – F\_E\_VX\_CL\_CARS
- Kniha jízd – F\_E\_VX\_CL\_LOG

Přehledové formuláře byly zaneseny do menu podnikového systému v podobě tří podadresářů – Evidence vozů, Kniha jízd, Přehled v hlavním adresáři „Kniha jízd“. Ve vývojovém prostředí byla těmto formulářům nastavena role master.

Obrázek 13 - Zobrazení knihy jízd v menu



Zdroj: Vlastní zpracování

Formulář F\_P\_VX\_CL\_CARS je přehledem evidovaných vozidel z tabulky Q\_VX\_CL\_CARS. Obsahuje sloupce SPZ, Název, Rok výroby, Průměrná spotřeba, Velikost nádrže, Zodpovědný pracovník. V řádku je také tlačítko Detail, které stejně jako Upravit záznam v panelu nástrojů po stisknutí otvírá editační formulář F\_E\_VX\_CL\_CARS.

Obrázek 14 - Přehledový formulář F\_P\_VX\_CL\_CARS

Přehled vozidel (Počet záznamů: 3)							v. 1.00
	SPZ	Název	Rok výroby	Průměrná spotřeba	Velikost nádrže	Zodpovědný pracovník	
1	test2	Toyota Corolla				Petra Andrejsková	
2	TEST	TEST2	2000			Petra Andrejsková	
3	5AN7415	Opel Zafira	2016	9	50	Martin Andreas	

Zdroj: Vlastní zpracování

Zde je sedm textových polí – ID (nevyplňuje se, automaticky se doplňuje pomocí sekvence z databáze, respektive za pomoci triggeru, který ji spouští) SPZ, Rok výroby, Název, Velikost nádrže, Průměrná spotřeba a Zodpovědný pracovník. Poslední jmenované pole lze vybrat pouze z předem daných možností. Přesněji příkazem SELECT z tabulky USERS.

Data, která se ve formuláři nacházejí, pochází z tabulky VX\_CL\_CARS, kde VX\_CL\_CARS.ID se rovná poli ve formuláři VX\_CL\_CARS.ID, kam se předává parametr z formuláře F\_P\_VX\_CL\_CARS, podle toho, na jakém řádku je stisknuté tlačítko Detail/Upravit záznam. Pokud je tento parametr nulový, uživatel stiskl tlačítko Nový záznam a pole ve formuláři F\_E\_VX\_CL\_CARS jsou prázdná.

Obrázek 15 - Editační formulář F\_E\_VX\_CL\_CARS

Detail vozidla

v. 1.00

Detail vozidla

ID

SPZ  Rok výroby

Název

Velikost nádrže  Průměrná spotřeba

Zodpovědný pracovník

Zdroj: Vlastní zpracování

Formulář F\_P\_VX\_CL\_LOG nad přehledem obsahuje filtrační pole: Vozidlo, Rok, Měsíc. U filtračního pole Vozidlo je omezení. V případě oprávnění Writer se zobrazují jen vozidla, u kterých je uživatel oprávněnou osobou. V případě oprávnění Developer uvidí uživatel všechna vozidla. S oprávněním Reader autorka zatím nepočítá, může být doplněno v další verzi knihy jízd. Filtrační pole Měsíc nabízí výběr z hodnot 1-12, v případě pole Rok je defaultně předvyplněné aktuálním rokem.

V přehledu se zobrazí odfiltrovaný seznam s informacemi. Obsahuje sloupce: Datum, Čas (start – cíl), Popis, Vzdálenost, Typ jízdy, Účel jízdy a tlačítko Detail, které stejně jako tlačítko Upravit záznam v panelu nástrojů po stisknutí otvírají editační formulář F\_E\_VX\_CL\_LOG.

Obrázek 16 - Přehledový formulář F\_P\_VX\_CL\_LOG

Přehled jízd

v. 1.01

Vozidlo  Rok  Měsíc

	Datum	Čas (start - cíl)	Popis	Vzdálenost	Typ jízdy	Účel jízdy
				3530 Σ		
1	07.12.2017	11:00 - 12:00	refresh	100	Pracovní	Jiný
2	07.12.2017	15:00 - 15:30		2555	Služební	
3	07.12.2017	16:00 - 18:00		500	Služební	Jiný
4	13.11.2017	-		25	Služební	Jiný
5	19.05.2017	-	T	100	Pracovní	
6	09.05.2017	14:00 - 18:00	Test - servisní jízda	250	Pracovní	Pracovní

Zdroj: Vlastní zpracování

Editační formulář zobrazuje dvě textová pole jen pro prohlížení a lepší přehlednost – Zodpovědný pracovník, SPZ z tabulky QVX\_CL\_CARS, kde VX\_CL\_CARS.ID je rovno poli VX\_CL\_LOG.ID\_CAR. Editací pole jsou Předchozí jízdy, Projekt, Účel jízdy, Typ jízdy, Datum, Čas stratu, Čas cíle, Vzdálenost, Popis. Pod těmito poli je záložka Čerpání obsahující tabulku VX\_CL\_FUEL\_LOG, kde VX\_CL\_FUEL\_LOG.ID\_LOG je rovno poli VX\_CL\_LOG.ID. Zde je možné vytvořit nebo smazat záznam pomocí tlačítek „Nový“ a „Smaž“. Data se ukládají stiskem ikony pro disketu v panelu nástrojů. Napravo od této ikonky se nachází tlačítko pro vrácení úprav.

Obrázek 17 - Editací formulář F E VX CL LOG

Detail jízdy

Zodpovědný pracovník: Bc. Petra Andrejsková  
 SPZ: test2  
 Projekt:   
 Účel jízdy:   
 Typ jízdy: Soukromé  
 Datum: 09.05.2017  
 Čas startu: 14:00  
 Čas cíle: 18:00  
 Vzdálenost: 250  
 Popis: TESTTEST

**Čerpání**

	Množství paliva	Cena	Stav tachometru	Poznámka	
	150 Σ	1500 Σ			
1	50	1500			
2	100				

Zdroj: Vlastní zpracování

Formulář přehled jízd F\_P\_VX\_CL\_OVERVIEW má podobnou vizuální podobu jako formulář knihy jízd, s tím rozdílem, že je rozšířený o pět sloupců pocházející z tabulky „Čerpání“ v editačním formuláři F\_E\_VX\_CL\_LOG a není zde zatím žádné omezení pro zobrazení záznamů. Autorka počítá s přiřazením speciální role v produkčním prostředí knihy jízd, která bude zaručovat přístup jen zaměstnancům finančního oddělení.

Obrázek 18 - Přehledový formulář F\_P\_VX\_CL\_OVERVIEW

Přehled jízd

Vozidlo:  Rok:  Měsíc:

☺	Datum	Čas (start - cíl)	Popis	Vzdálenost	Typ jízdy	Účel jízdy	Vlastník	SPZ	Vozidlo	Cena	Množství paliva
1	07.12.2017	11:00 - 12:00	refresh	100	Pracovní	Jiný	Bc. Petra Andrejsková	test2	Toyota Corolla		
2	07.12.2017	15:00 - 15:30		2555	Služební		Bc. Petra Andrejsková	test2	Toyota Corolla		
3	07.12.2017	16:00 - 18:00		500	Služební	Jiný	Bc. Petra Andrejsková	test2	Toyota Corolla		
4	07.12.2017	9:00 - 11:00	Bratislava	120	Služební	Pracovní	Martin Andreas	5AN7415	Opel Zafira		
5	05.12.2017	14:00 - 15:00	Bratislava	200	Služební	Pracovní	Martin Andreas	5AN7415	Opel Zafira		

Zdroj: Vlastní zpracování

### 4.5.3 Programování

Každý formulář má svou package v databázi, která je stejně pojmenována jako formulář. Zpravidla mají package dvě části: specifikaci a tělo. Ve specifikaci je přehled všech použitých procedur a funkcí, ty mají parametr ParAxForm in out nocopy PAX\_FORM.TYPE\_ITEMS, popřípadě další vstupní či výstupní parametr. V těle se nachází jejich samotná definice.

#### 4.5.3.1 Procedura CONTROL\_CAR

Tato procedura se volá ihned po spuštění formuláře F\_P\_VX\_CL\_LOG a udává, co bude koncovému uživateli zobrazeno za možnosti při výběru dat, konkrétně vozidla. Zjišťuje, jakou má přihlášený uživatel pravomoc v roli k prohlížení a ve volbě dat. Jinak se chová v případě role Developer nebo Writer. S rolí Reader se zatím nepočítá, a proto není v proceduře řešena.

U jednoho uživatele může být přiřazeno více vozidel. Z tohoto důvodu po splnění podmínky je zvolen kurzor k výběru dat. Na konci procedury se volá další LOAD\_DATE, která filtruje tabulku podle zvoleného roku a měsíce.

Ukázka procedury, která se spouští při výběru vozidla ve formuláři F\_P\_VX\_CL\_LOG:

```

PROCEDURE control_car (paraxform IN OUT NOCOPY pax_forms.type_items) AS
  pid NUMBER;
  mtableid VARCHAR2(100);
  pname VARCHAR2(256);
BEGIN
  IF instr(pax_users.getauthorization('T',pax_page.getcurtransaction(),NULL),'D') > 0 THEN
    DECLARE
      CURSOR c IS SELECT
        id,
        long_name
      FROM

```



```

        vx_cl_cars
    ORDER BY
        long_name;
    r c%rowtype;
BEGIN
    pax_forms_utl.removeitemlist(paraxform,'CONTROL.CAR');
    FOR r IN c LOOP
        pax_forms_utl.additemlist(paraxform,'CONTROL.CAR',r.ID,r.long_name,r.ID);
    END LOOP;
END;
END IF;

IF instr(pax_users.getauthorization('T',pax_page.getcurtransaction(),NULL),'W') > 0 THEN
    DECLARE
        CURSOR c IS SELECT
            id,
            long_name
        FROM
            vx_cl_cars
        WHERE
            id_user = pax_users.getloggedid
        ORDER BY
            long_name;
    r c%rowtype;
BEGIN
    pax_forms_utl.removeitemlist(paraxform,'CONTROL.CAR');
    FOR r IN c LOOP
        pax_forms_utl.additemlist(paraxform,'CONTROL.CAR',r.ID,r.long_name,r.ID);
    END LOOP;
END;
END IF;
-- if instr(pax_users.getauthorization('T', pax_page.getcurtransaction(), null),'R')>0 then
load_date(paraxform);
END control_car;

```

#### 4.5.3.2 Procedura LOAD\_DATE

Tato procedura se volá nejen v proceduře CONTROL\_CAR, ale i po výběru možností v poli „Měsíc“ nebo po zadání roku. Defaultně je nastaven stávající rok pomocí funkce sysdate. V případě „Měsíce“ je pole nevyplněné. Zde jsou ošetřeny možnosti, které mohou nastat: neplněný pole, měsíc, rok, popřípadě i vozidlo.

Ošetření filtrování formuláře po výběru vozidla:

```

PROCEDURE load_date (paraxform IN OUT NOCOPY pax_forms.type_items) AS
    mtableid VARCHAR2(256);
    sum_dist NUMBER;

```

```

qw VARCHAR(2000);
qw2 VARCHAR(2000);
pcar VARCHAR(256);
p_year VARCHAR(4);
p_month VARCHAR(2);
curr_year VARCHAR(4);
BEGIN
SELECT TO_CHAR(SYSDATE,'YYYY') INTO curr_year FROM dual;
p_year := paraxform('CONTROL.DATE_FROM').val;
p_month := paraxform('CONTROL.DATE_TO').val;
pcar := paraxform('CONTROL.CAR').val;
IF p_year IS NOT NULL AND p_month = 00 THEN
  qw := 'to_char(LOG_DATE,"YYYY") = ' || p_year || ' ';
ELSIF p_year IS NULL THEN
  qw := 'to_char(LOG_DATE,"YYYY") <= ' || curr_year || ' ';
ELSE
  qw := 'to_char(LOG_DATE,"YYYY") = ' || p_year
    || ' AND to_char(LOG_DATE,"MM") = ' || p_month || ' ';
END IF;

IF paraxform('CONTROL.CAR').val IS NOT NULL THEN
  qw2 := 'id_car = ' || pcar;
ELSE
  qw2 := 'id_car = 0';
END IF;

--ZAFILTROVÁNÍ TABULKY
mtableid := pax_axtable.getcomponentid(paraxform,'VX_CL_LOG');
pax_axtable.setwhereclause(mtableid,qw || ' AND ' || qw2);
END load_date;

```

#### 4.5.3.3 Procedura KEY\_COMMIT

Tlačítkem uložit ve formulář F\_P\_CL\_LOG se spouští procedura KEY\_COMMIT, která podmínkami zjišťuje, jestli je předaný parametr z přehledového formuláře prázdný a je vyplněné datum. Pokud jsou podmínky splněny, vkládá do databázové tabulky nové ID, respektive se spouští trigger, který ho vygeneruje. Poté vrací hodnotu ID do formulářového pole VX\_CL\_LOG.ID. Následně získá název aktuální komponenty a nastavuje serverový parametr na nové ID.

V případě, že je první podmínka nesplněna, jedná se o již založený záznam a ukládají se jen položky, které byly pozměněny. Pokud není splněna podmínka, tedy je nevyplněno datum, je uživatel na tuto událost upozorněn, a pokud chce nový záznam uložit, musí požadované pole vyplnit.

```

PROCEDURE key_commit (paraxform IN OUT NOCOPY pax_forms.type_items) AS
BEGIN
  IF paraxform('VX_CL_LOG.ID').val IS NULL THEN
    IF paraxform('VX_CL_LOG.LOG_DATE').val IS NULL THEN
      pax_forms.dialogok(paraxform,'Není zadané datum.','Chyba');
      return;
    ELSE
      paraxform('VX_CL_LOG.ID').val := pax_forms.axinsert(paraxform,'VX_CL_LOG.ID');
      COMMIT;
      pax_page.setserverpar(pax_forms_utl.getcomponentid(paraxform),'par_id',paraxform('VX_
      CL_LOG.ID').val);
    END IF;
  ELSE
    pax_forms.axupdate(paraxform);
    COMMIT;
  END IF;
  pax_forms.setinactive(paraxform);
END key_commit;

```

#### 4.5.3.4 Procedury BUTT\_NEW a BUTT\_DEL

Procedura BUTT\_NEW se spustí po stisknutí tlačítka Nový ve formuláři F\_E\_CL\_LOG. Do vytvořených proměnných předá ID vozidla a ID tabulky VX\_CL\_LOG a zakládá nový řádek v tabulce Čerpání. Hodnoty proměnných jsou poté vloženy do sloupců, pro které jsou určeny. Procedura je ukončena příkazem COMMIT pro uložení změn v tabulce.

V případě vymazání řádku z tabulky Čerpání slouží uživateli tlačítko Smaž, které volá proceduru BUTT\_DEL. Zde je jeden ze základních příkazů k vymazání řádku z databázové tabulky VX\_CL\_FUEL\_LOG. Procedura má parametr ID řádku a podle něj se vymaže požadovaný řádek.

```

PROCEDURE BUTT_NEW (paraxform IN OUT NOCOPY PAX_FORMS.TYPE_ITEMS) AS
id_car VARCHAR2(3);
pid VARCHAR2(3);
BEGIN
id_car := paraxform('VX_CL_LOG.ID_CAR').val;
pid := paraxform('VX_CL_LOG.ID').val;
INSERT
INTO VX_CL_FUEL_LOG
  (
    ID,
    ID_CAR,
    ID_LOG,
    AMOUT,
    PRICE,

```

```

MILEAGE,
LOG_NOTE
)
VALUES
(
null,
id_car,
pid,
null,
null,
null,
null
);
COMMIT;
END BUTT_NEW;

```

```

PROCEDURE BUTT_DEL (paraxform IN OUT NOCOPY PAX_FORMS.TYPE_ITEMS, prow rowid) AS
BEGIN
    DELETE FROM VX_CL_FUEL_LOG WHERE rowid=prow;
END;

```

#### 4.5.3.5 Procedura LOG\_USE

Procedura LOG\_USE je volána při výběru účelu jízdy ve formuláři F\_E\_VX\_CL\_LOG. Zde je aplikován explicitní kurz, který vrátí víc než jeden řádek z tabulky CODEBOOK do seznamu položek Účel jízdy, z kterých uživatel může vybírat hodnoty.

```

PROCEDURE LOG_USE (paraxform IN OUT NOCOPY PAX_FORMS.TYPE_ITEMS) AS
BEGIN
    DECLARE
        CURSOR c IS select short_name,
            long_name
        FROM vx_codebook
        WHERE codebook_type = 'CL_LOG_USE'
        ORDER BY long_name;
    r c%ROWTYPE;
BEGIN
    pax_forms_utl.removeitemlist(paraxform, 'VX_CL_LOG.LOG_USE');
    FOR r IN c LOOP
        pax_forms_utl.additemlist(paraxform, 'VX_CL_LOG.LOG_USE', r.short_name, r.long_name,
r.short_name);
    END LOOP;
END;
END LOG_USE;

```

#### 4.5.3.6 Procedura DETAIL

Tato procedura se nachází jak v package těla F\_E\_VX\_CL\_LOG, F\_P\_VX\_CL\_LOG, tak i ve stejnojmenných formulářích, kde je spuštěna při stisku tlačítka Detail nebo Editace v panelu nástrojů. Jejím vstupním parametrem je proměnná prow, která dostává ID řádku, ve kterém uživatel zrovna stiskl tlačítko. Jestliže dojde k stisknutí tlačítka a není vybrán žádný záznam, uživatel je na tuto skutečnost upozorněn systémovým hlášením. Po otevření editačního formuláře je záznam zamčen, tzn. že ho nemůžou upravovat ostatní uživatelé do té doby, pokud je okno zavřeno. Tato procedura je také použita při založení nových záznamů po stisku tlačítka Nový, kdy předávaný parametr PAR\_ID je nulový a editační formuláře se otvírají prázdné.

```
PROCEDURE DETAIL (paraxform IN OUT NOCOPY PAX_FORMS.TYPE_ITEMS, prow rowid,
preadonly IN varchar2 default 'false') AS
  newcompid VARCHAR2(256);
  locktable VARCHAR2(256) := 'VX_CL_CARS';
  editformname VARCHAR2(256) := 'F_E_VX_CL_CARS';
  pid NUMBER;
  vchyba VARCHAR2(20000);
  parreadonly VARCHAR2(5) := preadonly;
BEGIN
  axlogdebug('Procedure DETAIL started. ');
  SELECT id INTO pid FROM VX_CL_CARS WHERE rowid = prow;
  IF pid IS null THEN
    pax_forms.dialogok(paraxform,PAX_LANG.getmsg('Vyberte
záznam','CZ'),PAX_LANG.getmsg('Chyba','CZ'),2);
    RETURN;
  END IF;

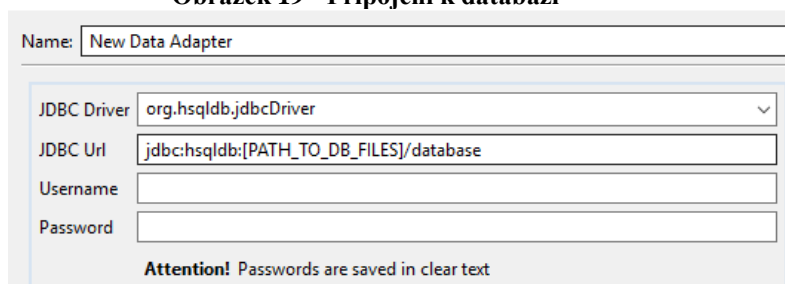
  --zamknu si objekt
  IF parreadonly = 'false' THEN
    IF not pax_forms_win.lockobject (locktable,pid,vchyba) THEN
      parreadonly := 'true';
      pax_forms.dialogok(paraxform,vChyba,pax_lang.getmsg('Informace','CZ'));
    END IF;
  END IF;

  --založím okno
  newcompid:=pax_forms_win.createformwindow(editformname,pax_lang.getmsg('Detail
vozidla','CZ'),'true');
  pax_forms_win.addformparameter(newcompid,'PAR_ID',pid);
  pax_forms_win.addformparameter(newcompid,'PAR_READ_ONLY', parreadonly);
  axlogdebug('Procedure DETAIL ended. ');
END DETAIL;
```

#### 4.5.4 Report

Aby report mohl být vytvořen, muselo nejprve dojít k propojení firemní databáze, která je zdrojem dat a vyplňuje report. To zahrnovalo zvolením JDBC Connection a vyplnění potřebných parametrů pro připojení, v tomto případě se jednalo o databázi Oracle. Samozřejmě lze zvolit jinou databázi podle toho, s jakou se pracuje. Následně byl definován vzhled dokumentu, včetně rozložení prvků, parametrů a proměnných, které reprezentují data.

Obrázek 19 - Připojení k databázi



Zdroj: Vlastní zpracování

Za pomoci SQL příkazů a tlačítka Read Fields se automaticky vygenerovaly sloupce z propojené databáze. Do hlavního přehledu přidala autorka nesystémové parametry (IDCAR, P\_YEAR, P\_MONTH), které jsou použity jako vstupní proměnné, přes které se dynamicky filtruje výběr příkazu SQL z databázové tabulky VX\_CL\_CARS. V reportu se nachází vnořená tabulka. Zde jsou přidány pomocné proměnné výpočet kilometrů za různých podmínek (SUM\_DISTANCE, SUM\_DISTANCE\_B, SUM\_DISTANCE\_P, END\_SUM\_KM) a parametry (TD\_P\_ID\_CAR, TD\_P\_YEAR, TD\_P\_MONTH, TD\_SUM\_TOTAL\_KM\_BEGIN), které mají vazbu na parametr z hlavního dotazu. Data, která tabulku plní, vycházejí z SQL příkazu z databázové tabulky VX\_CL\_LOG a stejného datového nastavení jako má hlavní report. V dialogovém editoru dotazu Jaspersoft neplatí pravidlo pro standardní ukončení SQL dotazu, proto se zde nenachází středník.

Příkaz pro propojení databáze s hlavním přehledem reportu:

#### SELECT

```
ID,  
ID_user,  
long_name,  
reg_plate,  
NVL((SELECT SUM(distance) FROM vx_cl_log WHERE ID_car = ${IDCAR}  
AND TO_CHAR(log_date,'YYYY') < ${P_YEAR}),0) AS top_km_begin_year,  
NVL((SELECT SUM(distance) FROM vx_cl_log WHERE ID_car = ${IDCAR}  
AND TO_CHAR(log_date,'YYYY') <= ${P_YEAR})
```

```

AND TO_CHAR (log_date,'MM') < $P{P_MONTH}),0) AS sum_total_km_begin,
    (SELECT
personal_number FROM vx_contacts WHERE ID_users = ID_user) AS person_ID,
    (SELECT
first_name||' '||last_name FROM ax_users WHERE ID = ID_user) AS person_name
FROM
    vx_cl_cars
WHERE ID=$P{IDCAR}

```

Příkaz pro vnořenou tabulku v reportu:

```

SELECT
    ID,
    ID_CAR,
    DECODE (B_P,'P',' ', TO_CHAR(LOG_DATE,'DD. MM. YYYY')) AS form_date,
    log_note,
    start_time,
    end_time,
    log_use,
    DECODE(LOG_USE, 'PRAC', 'PRACOVNÍ', 'JINY','JINÝ', 'OBCHOD', 'OBCHODNÍ', 'SERVIS',
'SERVIS VOZU') AS log_use_txt,
    distance,
    DECODE (B_P, 'P', 0, DISTANCE) AS b_distance,
    DECODE (B_P, 'P', DISTANCE, 0) AS p_distance,
    b_p
FROM vx_cl_log
WHERE ID_car = $P{TD_P_ID_CAR}
AND TO_CHAR(log_date,'YYYY') = $P{TD_P_YEAR}
AND TO_CHAR (log_date,'MM') = $P{TD_P_MONTH}
ORDER BY b_p ASC, log_date, start_time ASC

```

Po grafické stránce je reportu rozložen do čtyř samostatných bloků. V první a poslední části, tedy titulku a zápatí se nacházejí převážně prvky označované jako Static Fields. Jde o statický text, který zůstává neměnný při generování nové sestavy. Střední blok je zajímavý svým rozdělením na dvě části, kdy každá z nich vychází z jiných databázových dotazů, jak už bylo napsáno výše. Výsledky dotazů se generují do polí Text Field s vnitřním označením \$F {jméno\_sloupce}, \$P {jméno\_parametru}, \$V {jméno\_proměnné}, které jsou určeny pro dynamické zobrazování textu.

Po dokončení úprav a definování příkazů bylo třeba dát pozor na definování znakové sady u výstupního formátu PDF. Ta je defaultně nastavena na 1252 a plně nezobrazuje české znaky. Z toho důvodu se nastavuje znaková sada 1250, a to u všech tabulek a polí v reportu. Tomuto problému lze předejít nastavením vlastního stylu při zakládání nového reportu.

Report se generuje při stisknutí tlačítka „Tisk“ z databáze jako dokument ve formátu PDF v podobě pro tisk A4. Tlačítko zavolá proceduru z interní package formuláře

F\_P\_VX\_CL\_LOG. Pro získání reportu je nutné volat další pomocnou funkci, jejíž parametry jsou ID reportu a proměnné, které jsou vstupními hodnotami. Report je definován v tabulce REPORTS, kde ve sloupci CLOB je uložena šablona reportu ve formátu XML.

Procedura KEY\_PRINT pro volání reportu po stisknutí tlačítka „Tisk“:

```
PROCEDURE key_print (paraxform in out nocopy PAX_FORMS.TYPE_ITEMS) AS
  l_par pax_reports.t_rep_param;
  IDcar NUMBER := paraxform('CONTROL.CAR').Val;
  p_year NUMBER := paraxform('CONTROL.DATE_FROM').Val;
  p_month VARCHAR2 (10) := paraxform('CONTROL.DATE_TO').Val;
BEGIN
  IF p_month = 00 OR p_year is null THEN
    pax_forms.dialogok(paraxform, 'Není vybráný konkrétní měsíc.','Chyba');
  ELSE
    l_par('IDCAR'):= IDcar;
    l_par('P_YEAR'):= p_year;
    l_par('P_MONTH'):= p_month;
    paraxform ('TOOLBAR.KEY_PRINT').Val := pax_reports.getreportj(28, l_par);
  END IF;
END key_print;
```



Obrázek 20 - Grafický návrh reportu knihy jízd

<b>Kniha jízd firemního vozidla</b>						
Title						
Vazba na dokument: S0103		Garant formuláře: Petra Andrejsková			Označení: F0103 02	
Detail 1						
Jméno řidiče: <b>\$F{PERSON_NAME}</b>		SPZ: <b>\$F{REG_PLATE}</b>		Typ vozidla: <b>\$F{LONG_NAME}</b>		
		Rok: <b>\$P</b>	Měsíc: <b>\$P{P_MONTH}.equals</b>			
		Osobní číslo řidiče: <b>\$F{PERSON_ID}</b>				
Ujeto km						
Datum	Místo počátku - cíl - místo ukončení cesty	Od	Do	Služebně	Soukromě	Účel jízdy
<b>\$F{FORM_DATE}</b>	<b>\$F{LOG_NOTE}</b>	<b>\$F{START}</b>	<b>\$F{END_T}</b>	<b>\$F{B_DISTAN}</b>	<b>\$F{P_DISTA}</b>	<b>\$F{LOG_USE_TXT}</b>
	<b>Celkem</b>			<b>\$V</b>	<b>\$V</b>	
		<b>\$P</b>		<b>\$V{sum_distance}</b>		<b>\$V{end_sum_km}</b>
Detail 2						
Poznámky:		Podpis řidiče:		Schválit		
		Datum:		Datum:		
Účel služební jízdy:	Pracovní Obchodní	Jiný Sevis	"Stav ujetých km na počátku roku (zařazení do provozu): " + <b>\$F{TOP_KM_BEGIN_YEAR}</b> + " + "			
Summary						

Zdroj: Vlastní zpracování

## 4.5.5 Uživatelská příručka

Posledním krokem implementování knihy jízd je tvorba uživatelské příručky k formulářům F\_P\_VX\_CL\_LOG a F\_P\_VX\_CL\_CAR pro koncové uživatele. Obsahuje písemný návod k ovládní formulářů.

Uživatel si příručku stáhne v textové podobě ve formátu PDF po stisknutí tlačítka „Nápověda“ v panelu nástrojů. Jako příklad uvádí autorka uživatelskou příručku pro formulář F\_P\_VX\_CL\_LOG.

### 4.5.5.1 Obecné ovládní

#### 4.5.5.1.1 Panel nástrojů

Každý formulář obsahuje panel nástrojů

Obrázek 21 - Panel nástrojů





Zdroj: Vlastní zpracování

#### Význam tlačítek

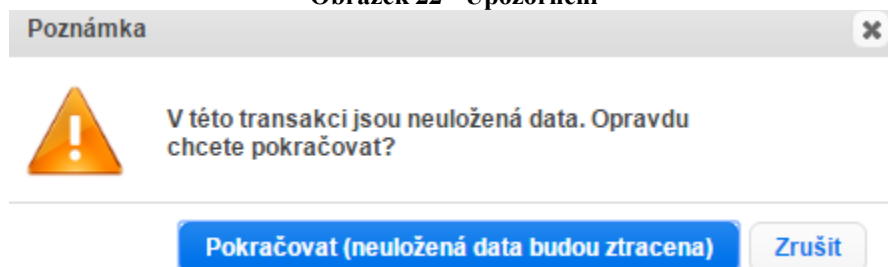
- Nový záznam (jen pro přehledy)
- Editace (zobrazení detailu)
- Odstranění záznamu
- Uložení (jen pro editaci)
- Vrátit zpět (jen pro editaci)
- Nápověda
- Zobrazení historie změn
- Tisk
- Export do Excelu (jen pro přehledy)
- Výběr zobrazení sloupců (jen pro přehledy)

### 4.5.5.2 Formuláře

Formuláře jsou buď přehledové, nebo editační. Přehledový formulář obsahuje panel nástrojů, tabulku s přehledem údajů a případně další akční tlačítka. Editační formulář se ve většině případů zobrazí jako okno s údaji k vyplnění/zobrazení. Okno lze zavřít  nebo klávesou Esc. Formulář obsahuje i záložky, kliknutím na záložku se zobrazí údaje z dané

záložky. Data se ukládají stiskem ikony  pro uložení v panelu nástrojů. Při uzavření okna zkontroluje systém, zda existují nějaké neuložené změny. Pokud ano, zeptá se:

Obrázek 22 - Upozornění



Zdroj: Vlastní zpracování

Jestliže zvolíte Pokračovat, systém data neuloží a formulář uzavře. Zvolíte tlačítko Zrušit, vrátíte se do formuláře a můžete data uložit.


#### 4.5.5.2.1 Přehled formuláře

Obrázek 23 - Přehledový formulář

	Datum	Čas (start - cíl)	Popis	Vzdálenost	Typ jízdy	Účel jízdy
				350 Σ		
1	09.05.2017	14:00 - 18:00	TESTTEST	250	Soukromé	
2	19.05.2017	-	T	100	Soukromé	



Zdroj: Vlastní zpracování

Přehled jízd se standardně ovládá pomocí panelu nástrojů (viz kapitola Panel nástrojů). U kolonky „Vozidlo“ vybereme ze seznamu vozidlo, u kterého chceme zobrazit seznam jízd. Pokud chceme ještě upravit výběr určitého data, zadáme námi zvolené datum do kolonky „Od“ a „Do“.

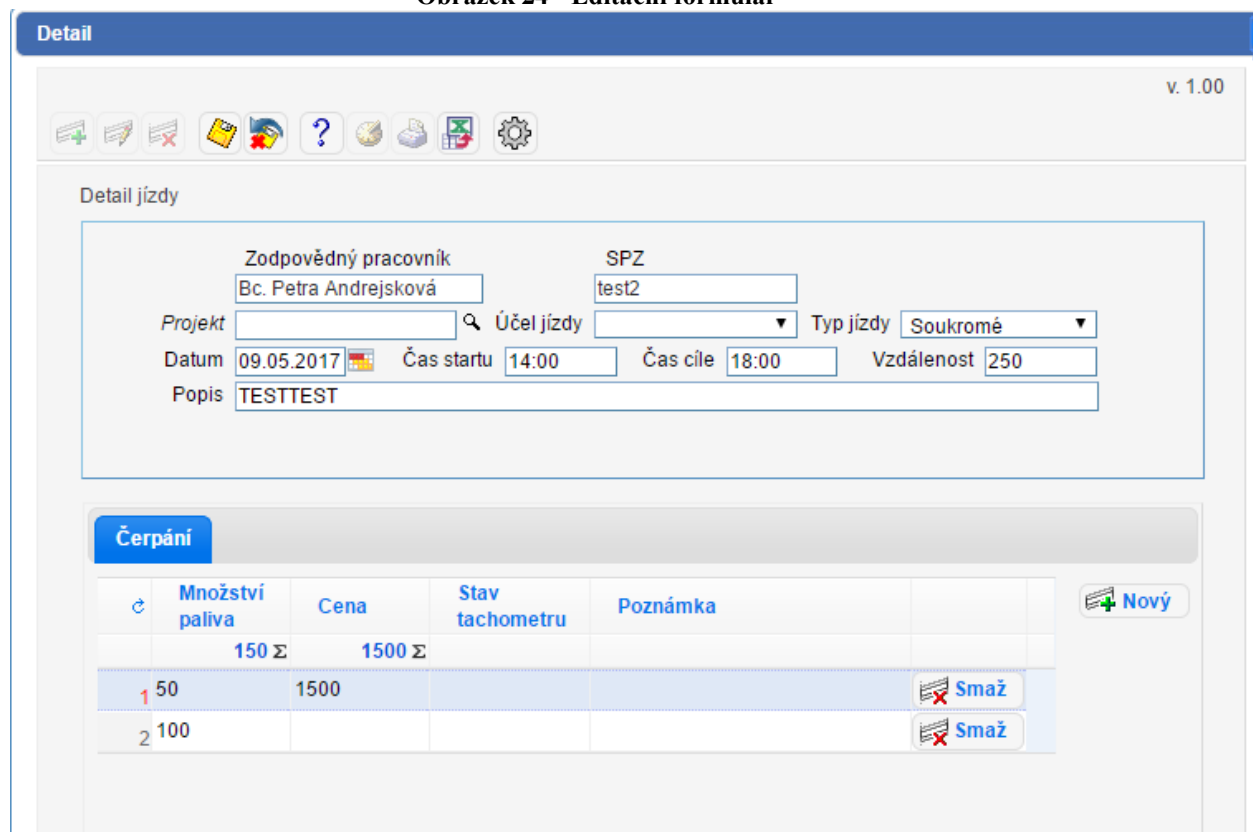
Tlačítkem  zobrazíte detail jízdy, který můžete editovat. To samé uděláte i



tlačítkem  v panelu nástrojů.

#### 4.5.5.2.2 Detail jízdy




Kliknutím na tlačítko  v panelu nástrojů či na tlačítko  v řádku formuláře u daného záznamu zobrazíte tuto tabulku:

Obrázek 24 - Editační formulář





	Množství paliva	Cena	Stav tachometru	Poznámka	
	150 Σ	1500 Σ			
1	50	1500			
2	100				

Zdroj: Vlastní zpracování

Tabulku nelze editovat, pokud je jen pro čtení. Pokud tomu tak není, lze upravovat „Projekt“ kliknutím na tlačítko , účel jízdy (výběr ze seznamu), typ jízdy (výběr ze seznamu), datum, čas startu a cíle, vzdálenost (km), popis (např. název tankovací stanice). Kolonky „Zodpovědný pracovník“ a „SPZ“ nelze upravovat. Pod ohraničenou tabulkou je záložka Čerpání. Tu lze editovat přímo kliknutím do daného sloupce, mazat řádky tlačítkem  nebo přidat nový řádek tlačítkem .

#### 4.5.5.3 Založení nového záznamu

Pro založení nového záznamu použijte tlačítko  v přehledovém formuláři. Otevře se nevyplněný formulář se stejným vzhledem jako při kliknutí na tlačítko .


Obrázek 25 - Založení nového záznamu

The screenshot shows a software window titled "Nový záznam" (New Record) with a close button in the top right corner. The version number "v. 1.01" is displayed in the top right. A toolbar with various icons is located below the title bar. The main content area is titled "Detail jízdy" (Trip Detail) and contains a form with the following fields:

- Zodpovědný pracovník: Petra Andrejsková
- SPZ: test2
- Předchozí jízdy: [input field]
- Projekt: [input field]
- Účel jízdy: [dropdown menu]
- Typ jízdy: [dropdown menu]
- Datum: [input field]
- Čas startu: [input field]
- Čas cíle: [input field]
- Vzdálenost: [input field]
- Popis: [input field]

Below the form is a section titled "Čerpání" (Refueling) with a table. The table has the following columns: "Množství paliva" (Fuel quantity), "Cena" (Price), "Stav tachometru" (Speedometer status), and "Poznámka" (Note). A "Nový" (New) button is located to the right of the table.

Zdroj: Vlastní zpracování

Po vyplnění záznamů nezapomeňte uložit stisknutím tlačítka .

## 5 Výsledky a diskuze

V rámci revitalizace a lepšího přehledu o jednotlivých jízdách byla zaimplementována nová elektronická kniha jízd, která je tématem této diplomové práce. Umožňuje přehled o jízdách všech zaměstnanců za celý rok na jednom místě. Formuláře lze zafiltrovat podle aktuálních požadavků a popřípadě stáhnou v podobě PDF. (Grafická podoba staré verze knihy jízd byla použita pro vytvoření reportu.) Vyplňování knihy jízd je možné kdekoliv a odkudkoliv, kde je internetové připojení.

Kniha jízd je přístupná na interních firemních stránkách pro zaměstnance v záložce VUSTE – APIS v podadresáři Kniha jízd. Tento podadresář je rozdělen na tři části (záložky) – Evidence vozů, Kniha jízd, Přehled. Záložka – Evidence vozů – je přístupná jen pro uživatele s rolí master. Dále tato záložka umožňuje editovat či zakládat nový záznam. Zde jsou informace o firemním vozidle a je k ní přiřazen zodpovědný pracovník. Druhá část – Kniha jízd – nabízí založení nového záznamu, editaci a filtrování podle značky vozidla, roku a měsíce. Pod filtrovacími poli se nachází tabulka s přehledem jízd pro zvolené vozidlo.

Každému řádku v tabulce náleží tlačítko „Detail“. Zde je také omezení. Uživateli se zobrazí pouze vozidlo, ke kterému je přiřazen jako zodpovědný pracovník. Po rozkliknutí daného záznamu se otevře formulář s údaji o jízdě vozidla. Pod údaji je zobrazena tabulka týkající se čerpání pohonných hmot i s tlačítkem přidání nového záznamu.

V poslední záložce „Přehled“ jsou zobrazena ta samá pole pro filtrování jako v záložce „Kniha jízd“ s tím rozdílem, že zde zobrazuje všechna firemní vozidla. Přehled není určen pro editaci, ale jen pro čtení. Přístup k němu mají všichni uživatelé.

Důležitým aspektem v závěru řešeného problému bylo testování aplikace. Zde došlo k zjištění, že v reportu se nezobrazovala dynamická textové pole. Důvodem byla nezjištěná příčina nemožnosti generování reportu serverem font Sans Serif. Proto při dalších řešení je třeba dát pozor na správné nastavení fontu Arial Narrow u jednotlivých polí.

V důsledku rozvoje moderních technologií a také vzhledem k rostoucímu počtu vozidel, a to zejména registrovaných jako firemní, je logické, že dojde i k aktualizaci knihy jízd. Z předchozích kapitol je zřejmé, že navržení knihy jízd byl rozsáhlý projekt – od analýzy stávajícího řešení až po sepsání uživatelské příručky – ten nahradil předchozí řešení, které bylo vedené jen v Excelu, kde v jednotlivých sešitech byly tabulky pro každý měsíc.

Díky zpětné vazbě uživatelů, je zřejmé, že je řešení přívětivější, jak ve vizuální podobě, tak i v rychlosti vyplňování knihy jízd.

Praktická část diplomové práce představuje jedno z možných řešení knihy jízd využívané v menším nebo středním podniku. Demonstruje reálné využití relační databázové technologie a jazyků SQL a PL/SQL. Technické řešení je založeno zejména na databázi od firmy Oracle, která je jádrem celého systému. Zde jsou uložena veškerá data, která jsou zobrazena ve formulářích za pomoci webového prohlížeče. Kromě toho se zde nacházejí i jednotlivé packages s funkcemi a procedurami, díky jejich pomoci dochází k ovládní jednotlivých formulářů.

Lze diskutovat o variantách možností realizace knihy jízd:

- Varianta 1: Ponechání stávajícího, na jedné straně zavedeného, na druhé straně těžkopádného procesu vykazování služebních cest fyzickým předáním vyplněných excelovských tabulek vytištěných a podepsaných řidičem a schvalovatelem.
  - Výhody této varianty: nic se nemusí měnit.
  - Nevýhody této varianty: čas pro podpisy, ztrácení dokumentů, špatný přehled všech služebních cest a celkových nákladů na amortizaci vozidel ve firmě a čerpání pohonných hmot.
- Varianta 2: Ponecháním Excelu jako nástroje pro vyplnění údajů a přeposílání Excelu pomocí procesního schvalovacího modulu ve VAMAXu
  - Výhody: je možné implementovat makro pro všechny Excely na jednom úložišti a shrnutí všech služebních cest (náklady, ujeté kilometry) např. za jeden měsíc.
  - Nevýhody: Zůstává těžkopádnost se sháněním podpisů jako v případě varianty 1. Pro souhrnný seznam výsledků je nutné, aby stále některý ze zaměstnanců firmy kontroloval souhrnný tabulkový procesor.
- Varianta 3: Kniha jízd jakožto integrální součást VAMAXu
  - Výhody: poskytuje veškeré informace souhrnně od všech zaměstnanců na vyžádání, dále obsahuje registr vozů. Další výhody rozepsány níže.
  - Nevýhody: implementace není složitá, však časově náročnější než tvorba makra v Excelu ve variantě 2.

V neposlední řadě nespornou výhodou řešení knihy jízd v systému VAMAX je skutečnost, že společnost VUSTE-APIS vážně zvažuje prodej systému VAMAX jako

škálovatelné komplexní IT stavebnice, tedy jako procesní nástroj pro zákazníky v České republice a na Slovensku. Pokud budou obchodní vyjednávání se stávajícími nebo novými zákazníky úspěšná, mohou si zákazníci zvolit také Knihu jízd a Registr vozidel. Není třeba zdůrazňovat, že rozvoj IT není pouze doménou komerčních společností, nýbrž také státní správy, kde jednotný IT systém jakožto uživatelsky příjemná nadstavba databáze Oracle ušetří spoustu času a peněz spojených s interními procesy. Toto je patrně největší výhoda varianty VAMAX, která je popsána v této práci.

Je třeba dbát na neustálý životní cyklus aplikace a počítat s neustálými vylepšeními, která budou usnadňovat uživatelům práci a šetřit čas. Důležité je uživatele dostatečně zaškolit, aby se nebáli nové funkce, které by mohly být přidány, používat. V praxi společnosti VUSTE-APIS to znamená, že kniha jízd a registr vozidel budou navázány na finanční a auditní komponenty VAMAXu. Toto jsou další velmi důležité integrace, které dále usnadní chod firmy VUSTE-APIS. Účtárna bude mít k dispozici všechny náklady všech vozů autoparku poloautomaticky, to bude docíleno integrací systému VAMAX s účetním systémem Pohoda, tedy výstup modulu Knihy jízd z VAMAXu vygeneruje XML soubor, který si účetní načte do SW Pohoda. Pro auditní účely je také export knihy jízd do XML nebo PDF velmi příznivý.



## 6 Závěr

Cílem diplomové práce bylo vytvoření funkční knihy jízd do podnikového systému VAMAX ve společnosti VUSTE-APIS, s.r.o. za pomoci databázového systému Oracle v aplikaci Oracle SQL Developer.

Společnost VUSTE-APIS, s. r. o. je menší pražskou firmou s pětadesátiletou tradicí a přibližně s třiceti zaměstnanci, která se po roce 1989 postupně transformovala a privatizovala. V současné době je moderním subjektem, který se zabývá vývojem a implementací softwaru a nabízí softwarové produkty svým zákazníkům. Jedním z jejich produktů je i softwarový podnikový informační systém VAMAX.

Prvním dílčím krokem v procesu tvorby této diplomové práce byla orientace ve firemním systému a zmapování stávajícího stavu řešení vedení knihy jízd a provedení analýzy všech náležitostí databáze této knihy. Výhodou autorky tohoto textu je, že prostředí firmy dobře zná, protože tam již třetím rokem pracuje. Vytvoření knihy jízd bylo jejím prvním samostatným pracovním projektem, který se měl stát součástí systému VAMAX.

VA manager (VAMAX) je informační systém nové generace, který přispívá k efektivitě práce tím, že firmě šetří nejen čas, ale i peníze. Využívá vysoký technický standart zabezpečení a vzhledem k tomu je bezpečný. Může se spustit pomocí běžného internetového prohlížeče, nepotřebuje tedy žádný specializovaný program pro spuštění. Lze konstatovat, že funguje jako stavebnice, kde si uživatel volí služby, které potřebuje. Ten se pak může soustředit na svou práci a může ji sdílet i se svými spolupracovníky.

Součástí tohoto systému se stala plně funkční kniha jízd, která snížila administrativu při jejím vedení, ušetřila čas při vyplňování, zajistila lepší přehled a umožnila víceúrovňový systém pro přístup k záznamům. Je určena pro zaměstnance menší nebo střední firmy s přiděleným vozidlem, ale slouží také finančnímu oddělení a vedení firmy, aby bylo možné sledovat efektivitu finančních prostředků na čerpání pohonných hmot. Hlavní cíl práce byl tedy naplněn, neboť realizace SW řešení funkční knihy jízd v podnikovém systému VAMAX vede k větší transparentnosti, ke snížení administrativy, a tudíž k časovým a finančním úsporám.

Dílčí kroky při tvorbě práce vedly k navržení logického modelu, zejména pro lepší přehled propojení jednotlivých tabulek a eliminaci redundantních dat, vytvoření datového slovníku pro dodržení doménové a entitní integrity, ztvárnění přívětivého grafického prostředí. Zásadní částí bylo samotné programování v databázových jazycích SQL

a PL/SQL v aplikaci Oracle SQL Developer. To zahrnovalo tvorbu tabulek a jejich propojení pomocí SQL a vytvoření balíčku pro uložení procedur a funkcí prostřednictvím PL/SQL. Následně došlo k vizuálnímu propojení s podnikovým systémem VAMAX. Další fází práce byla realizace reportu v aplikaci Jaspersoft Studio, který generuje data ve formě PDF v takové formě, aby mohl být vytištěn ve formátu A4 a archivován v tištěné podobě. Posledním důležitým krokem implementování knihy jízd bylo vytvoření uživatelské příručky pro formuláře F\_P\_VX\_CL\_LOG a F\_P\_VX\_CL\_CAR, která slouží potenciálním uživatelům a obsahuje písemný návod k jejich ovládní.

Popsané předchozí kroky jsou součástí praktické části (kapitola 4 Vlastní práce), která je stěžejní částí této diplomové práce. K naplnění cíle bylo třeba nastudovat především relačně databázové technologie související s řešeným problémem (viz. Kapitola 3 Teoretická východiska). Autorka prošla procesem vyhledávání a studiem relevantní informačních zdrojů vhodných k zvolené problematice (důležitým zdrojem byla kniha Databázové systémy od autorů Pokorný, Valenta a dokumentaci firmy ORACLE). Dále provedla sumarizaci teoretických poznatků, ze kterých vychází tato část. Důležité podkapitoly se nachází v závěru této kapitoly a jsou věnovány databázovým dotazovacím jazykům SQL, PL/SQL a jejich struktuře, díky nim je řešena z velké části problematika knihy jízd. Cílem této kapitoly bylo seznámení s literární rešerší vycházející z odborné literatury a dokumentace firmy ORACLE, zaměřené na problematiku v praktické části.

Hlavním přínosem celé práce je zaimplementování nové knihy jízd do systému jako funkční samostatný modul, který je reálně nasazen v podniku VUSTE-APIS a využíván všemi jeho zaměstnanci. Postup jejího zpracování je předmětem praktické části, ta představuje jedno z možných řešení knihy jízd, jak může být využívána v menším či středním podniku a představuje využití databázové technologie a jazyků SQL a PL/SQL. V textu je diskutováno o variantách možností realizace knihy jízd, a to původního systému vykazování služebních cest v excelových tabulkách, nebo ponechání Excelu jako nástroje pro vyplňování v Excelu a přeposílání schvalovacího modulu ve VAMAXu. Jako nejvýhodnější se jeví kniha jízd, která je plně integrována do systému VAMAX, neboť se vykazuje lepší přehled jízd, kniha jízd poskytuje veškeré údaje od všech zaměstnanců souhrnně, tedy je možné zobrazit všechny jízdy za uplynulý rok u jakéhokoliv zaměstnance či zaměstnanců a finanční oddělení firmy může s těmi to informacemi lépe pracovat a efektivněji sledovat náklady na čerpání pohonných hmot. Vytvořený prototyp byl testován, před ostrým nasazením do provozu. V alfa testování nebyla shledána žádná chyba, v rámci

beta testování se objevily drobné problémy, a to zejména v nastavení fontu u jednotlivých polí v reportu. To způsobilo nesprávné vygenerování a následně nečitelnost reportu. Výše uvedený problém byl řešen změnou fontu. V rámci beta testování byla verze knihy jízd povýšena na 1.02, kdy bylo přidáno tlačítko pro dopočet ujetých kilometrů v soukromých jízdách.

V budoucnosti firma VUSTE-APIS počítá s dalším doplnění funkcí vzhledem k tomu, že chce produkt VAMAX, jehož součástí je integrovaná kniha jízd, nabídnout dalším uživatelům v České i Slovenské republice. A teprve trh ukáže možnosti dalšího vývoje a doplnění tohoto produktu. Není třeba zdůrazňovat, že zákazníci mohou být firmy z komerční oblasti, ale i ze státní správy, kde by jednotný informační systém přinesl úsporu času a peněz. Pokud budou uživatelé dostatečně a dobře zaškoleni, usnadní jim značně práci. Z výše uvedeného vyplývá, že navržené řešení je možné využívat v obdobných institucích jako je firma VUSTE-APIS.

Největší výhodou knihy jízd zaintegrované do systému VAMX je to, že jednotný IT systém, jež je příjemnou nadstavbou databáze Oracle, ušetří čas spojený s interními procesy, a tudíž i finanční prostředky. Je možné přiřadit speciální pole v produkčním prostředí knihy jízd, které bude umožňovat přístup jen zaměstnancům finančního oddělení. Autorka předpokládá, že v dalším vývojovém cyklu výsledné aplikace by mohla být ke každému vozidlu přidána možnost naplánování údržby, jako například STK nebo pravidelnou servisní prohlídku, a to podle počtu ujetých kilometrů s tím, že by byl uživatel systémem upozorněn na tuto skutečnost. Dalším návrhem by mohlo být zobrazování statistiky týkající se kvality jednotlivých řidičů s příslušnými ikonkami. Nabízí se také možnost přidání GPS systému do každého vozidla s možností nepřetržitého sledování jízd s mapovým podkladem. Autorka je přesvědčena, že bude firmou pověřena k dalšímu doplnění knihy jízd podle požadavků zákazníků.

## 7 Seznam použitých zdrojů

- (1) LACKO, Luboslav. *Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]*. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4.
- (2) *www.lucidchart.com* [online]. [cit. 2018-06-25]. Dostupné z: <https://www.lucidchart.com/pages/database-diagram/database-models>
- (3) *www.gitta.info* [online]. [cit. 2018-06-25]. Dostupné z: [http://www.gitta.info/DBSysConcept/en/html/DataModSchem\\_learningObject1.html](http://www.gitta.info/DBSysConcept/en/html/DataModSchem_learningObject1.html)
- (4) STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. *Naučte se SQL za 28 dní: [stačí hodina denně]*. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.
- (5) POKORNÝ, J., VALENTA, M.: *Databázové systémy*. ČVUT, Praha 2013. ISBN 978-80-01-05212-9
- (6) *my.vertica.com* [online]. [cit. 2018-06-25]. Dostupné z: <https://my.vertica.com/docs/8.1.x/HTML/index.htm#Authoring/Glossary/Cardinality.htm>
- (7) *informacni-technologie.studentske.cz* [online]. [cit. 2018-06-25]. Dostupné z: <http://informacni-technologie.studentske.cz/2009/02/kardinalita-vztahu.html>
- (8) *oracle.com.edgesuite.net* [online]. [cit. 2018-06-25]. Dostupné z: <http://oracle.com.edgesuite.net/timeline/oracle/>
- (9) *www.tutorialspoint.com* [online]. [cit. 2018-06-25]. Dostupné z: <https://www.tutorialspoint.com/sql/sql-overview.htm>
- (10) ABBEY, Michael, Ian ABRAMSON a Michael COREY. *Oracle9i: [základy práce s databází Oracle9i]*. Praha: SoftPress, 2002. ISBN 80-864-9724-0.
- (11) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/database/121/TGSQL/tgsql\\_sqlproc.htm#TGSQL176](https://docs.oracle.com/database/121/TGSQL/tgsql_sqlproc.htm#TGSQL176)
- (12) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/cd/E11882\\_01/server.112/e40540/sqlangu.htm#CNCPT173216](https://docs.oracle.com/cd/E11882_01/server.112/e40540/sqlangu.htm#CNCPT173216)
- (13) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_1001.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_1001.htm)
- (14) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_4010.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_4010.htm)
- (15) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/cd/E11882\\_01/server.112/e40540/sqlangu.htm#CNCPT516](https://docs.oracle.com/cd/E11882_01/server.112/e40540/sqlangu.htm#CNCPT516)
- (16) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/database/121/LNPLS/create\\_trigger.htm#LNPLS01374](https://docs.oracle.com/database/121/LNPLS/create_trigger.htm#LNPLS01374)
- (17) *matejcerny.cz* [online]. [cit. 2018-06-25]. Dostupné z: <http://matejcerny.cz/blog/plsql-v-prikladech-procedury-a-funkce/>

- (18) *docs.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: [https://docs.oracle.com/cd/E11882\\_01/appdev.112/e25519/controlstatements.htm#LNPLS00402](https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/controlstatements.htm#LNPLS00402)
- (19) *jls.webz.cz* [online]. [cit. 2018-06-25]. Dostupné z: <http://jls.webz.cz/mff/oracle/plsql.html>
- (20) MCLAUGHLIN, Michael. *Oracle database 11g PL/SQL programming*. New York: McGraw-Hill, c2008. ISBN 00-714-9445-6.
- (21) *www.oracle.com* [online]. [cit. 2018-06-25]. Dostupné z: <http://www.oracle.com/technetwork/developer-tools/sql-developer/what-is-sqldev-093866.html>

## 8 Přílohy

### (A) XML kód formulářů

```
<!--F_P_VX_CL_CARS-->
<Module xmlns="http://xmlns.oracle.com/Forms">
  <FormModule ConsoleWindow="WINDOW" Name="F_P_VX_CL_CARS">
    <Block Name="CONTROL">
      </Block>
      <Canvas Name="CANVAS" WindowName="WINDOW" Width="664" Height="223"
ViewportXPosition="1" ViewportYPosition="44"/>
      <Canvas Name="SYSCANVAS_TOP" Width="664" Height="40"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="1" ViewportWidth="undefined" ViewportHeight="undefined">
      </Canvas>
      <Canvas Name="SYSCANVAS_LOW" Width="664" Height="18"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="270" ViewportWidth="undefined" ViewportHeight="undefined">
      </Canvas>
      <Block RecordsDisplayCount="10" Name="VX_CL_CARS" Height="247"
Width="654" QueryDataSourceName="Q_VX_CL_CARS" DatabaseBlock="true"
InsertAllowed="false" UpdateAllowed="false">
        <Item Width="100" Height="18" Name="ID" ItemType="Text Item"
CanvasName="CANVAS" Visible="false" XPosition="4" YPosition="4"/>
        <Item Width="50" Height="16" Name="DETAIL" ItemType="Push Button"
CanvasName="CANVAS" Label="Detail" StyleClass="key_edit"/>
        <Item Width="70" Height="16" Name="REG_PLATE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="SPZ" Enabled="false"/>
        <Item Width="130" Height="16" Name="LONG_NAME" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Název" Enabled="false"/>
        <Item Width="70" Height="16" Name="YEAR_CONST" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Rok výroby" Enabled="false"/>
        <Item Width="70" Height="16" Name="GAS_MILEAGE" ItemType="Text
Item" CanvasName="CANVAS" Prompt="Průměrná<br> spotřeba"
Enabled="false"/>
        <Item Width="70" Height="16" Name="TANK_SIZE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Velikost <br>nádrže" Enabled="false"/>
        <Item Width="130" Height="16" Name="ID_USER" ItemType="Text Item"
CanvasName="CANVAS" Enabled="false" Visible="false"/>
        <Item xmlns="" Width="130" Height="18" Name="USER_NAME"
ItemType="Text Item" Prompt="Zodpovědný <br>pracovník"
CanvasName="CANVAS"/>
      </Block>
      <Block Name="SYSBLOCK">
        <Item YPosition="5" XPosition="631" Width="30" Height="18"
Name="VERZE" ItemType="HTML" CanvasName="SYSCANVAS_TOP" Value="v. 1.00"/>
        <Item YPosition="3" XPosition="5" Width="416" Height="18"
Name="TOPIC" ItemType="HTML" CanvasName="SYSCANVAS_TOP"/>
        <Item YPosition="19" XPosition="5" Width="20" Height="20"
Name="KEY_NEW" Enabled="true" ItemType="Push Button" CanvasName="SYSCANVAS_TOP"
StyleClass="key_new" Tooltip="Nový záznam"/>
        <Item YPosition="19" XPosition="25" Width="20" Height="20"
Name="KEY_EDIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_edit" Tooltip="Upravit záznam"/>
        <Item YPosition="19" XPosition="45" Width="20" Height="20"
Name="KEY_DELETE" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_delete" Tooltip="Smazat záznam"/>
      </Block>
    </FormModule>
  </Module>

```

```

        <Item YPosition="19" XPosition="70" Width="20" Height="20"
Name="KEY_COMMIT" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_commit" Tooltip="Uložit"/>
        <Item YPosition="19" XPosition="90" Width="20" Height="20"
Name="KEY_REFRESH" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_refresh" Tooltip="Obnovit data"/>
        <Item YPosition="19" XPosition="115" Width="20" Height="20"
Name="KEY_HELP" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_help" Tooltip="Nápověda"/>
        <Item YPosition="19" XPosition="135" Width="20" Height="20"
Name="KEY_HISTORY" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_history" Tooltip="Historie
záznamu"/>
        <Item YPosition="19" XPosition="155" Width="20" Height="20"
Name="KEY_PRINT" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_print" Tooltip="Tisk"/>
        <Item YPosition="19" XPosition="175" Width="20" Height="20"
Name="KEY_XLSX" Enabled="true" ItemType="File Download"
CanvasName="SYSCANVAS_TOP" StyleClass="key_xlsx" Tooltip="Export do excelu"
ItemProcedure="KEY_XLSX" ItemProcedureParam="VX_CL_CARS"/>
        <Item YPosition="19" XPosition="200" Width="20" Height="20"
Name="KEY_SETTINGS" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_settings" Tooltip="Nastavení"
ItemProcedureParam="VX_CL_CARS" ItemProcedure="KEY_SETTINGS"/>
        <Item YPosition="1" XPosition="1" Width="457" Height="18"
Name="STATUSBAR" ItemType="HTML" CanvasName="SYSCANVAS_LOW"/>
    </Block>
</FormModule>
</Module>

<!--F_E_VX_CL_CARS-->

<Module xmlns="http://xmlns.oracle.com/Forms">
    <FormModule ConsoleWindow="WINDOW" Name="F_E_VX_CL_CARS">
        <Block Name="CONTROL">
            <Item YPosition="12" XPosition="12" Width="100" Height="13"
Name="POP_CARS" ItemType="HTML" Value="Detail vozidla" CanvasName="CANVAS"/>
        </Block>
        <Canvas Name="CANVAS" WindowName="WINDOW" Width="366" Height="165"
ViewportXPosition="1" ViewportYPosition="43">
            <Graphics Name="RAM_CARS" XPosition="14" YPosition="24" Width="339"
Height="117"/>
        </Canvas>
        <Canvas Name="SYSCANVAS_TOP" Width="366" Height="40"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="0" ViewportWidth="undefined" ViewportHeight="undefined">
        </Canvas>
        <Canvas Name="SYSCANVAS_LOW" Width="366" Height="18"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="211" ViewportWidth="undefined" ViewportHeight="undefined">
        </Canvas>
        <Block Name="SYSBLOCK">
            <Item YPosition="2" XPosition="335" Width="30" Height="18"
Name="VERZE" ItemType="HTML" CanvasName="SYSCANVAS_TOP" Value="v. 1.00"/>
            <Item YPosition="3" XPosition="5" Width="311" Height="18"
Name="TOPIC" ItemType="HTML" CanvasName="SYSCANVAS_TOP"/>

```

```

        <Item YPosition="19" XPosition="5" Width="20" Height="20"
Name="KEY_NEW" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_new" Tooltip="Nový záznam"/>
        <Item YPosition="19" XPosition="25" Width="20" Height="20"
Name="KEY_EDIT" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_edit" Tooltip="Upravit záznam"/>
        <Item YPosition="19" XPosition="45" Width="20" Height="20"
Name="KEY_DELETE" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_delete" Tooltip="Smazat záznam"/>
        <Item YPosition="19" XPosition="70" Width="20" Height="20"
Name="KEY_COMMIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_commit" Tooltip="Uložit"/>
        <Item YPosition="19" XPosition="90" Width="20" Height="20"
Name="KEY_REFRESH" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_refresh" Tooltip="Obnovit data"/>
        <Item YPosition="19" XPosition="115" Width="20" Height="20"
Name="KEY_HELP" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_help" Tooltip="Nápověda"/>
        <Item YPosition="19" XPosition="135" Width="20" Height="20"
Name="KEY_HISTORY" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_history" Tooltip="Historie
záznamu"/>
        <Item YPosition="19" XPosition="155" Width="20" Height="20"
Name="KEY_PRINT" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_print" Tooltip="Tisk"/>
        <Item YPosition="19" XPosition="175" Width="20" Height="20"
Name="KEY_XLSX" Enabled="false" ItemType="File Download"
CanvasName="SYSCANVAS_TOP" StyleClass="key_xlsx" Tooltip="Export do excelu"
ItemProcedure="KEY_XLSX" ItemProcedureParam="QAXP_ROLES"/>
        <Item YPosition="19" XPosition="200" Width="20" Height="20"
Name="KEY_SETTINGS" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_settings" Tooltip="Nastavení"
ItemProcedureParam="QAXP_ROLES" ItemProcedure="KEY_SETTINGS"/>
        <Item YPosition="3" XPosition="10" Width="355" Height="18"
Name="STATUSBAR" ItemType="HTML" CanvasName="SYSCANVAS_LOW"/>
    </Block>
    <Block Name="VX_CL_CARS" WhereClause="ID=:VX_CL_CARS.ID"
QueryDataSourceName="VX_CL_CARS" UpdateAllowed="true">
        <Item XPosition="108" YPosition="41" Width="37" Height="14"
Name="ID" ItemType="Text Item" CanvasName="CANVAS" Enabled="false" Prompt="ID"
PromptAttachmentOffset="5"/>
        <Item XPosition="108" YPosition="56" Width="100" Height="14"
Name="REG_PLATE" ItemType="Text Item" CanvasName="CANVAS" Prompt="SPZ"
PromptAttachmentOffset="5"/>
        <Item XPosition="268" YPosition="56" Width="40" Height="14"
Name="YEAR_CONST" ItemType="Text Item" CanvasName="CANVAS" Prompt="Rok výroby"
PromptAttachmentOffset="5"/>
        <Item XPosition="108" YPosition="71" Width="200" Height="14"
Name="LONG_NAME" ItemType="Text Item" CanvasName="CANVAS" Prompt="Název"
PromptAttachmentOffset="5"/>
        <Item XPosition="108" YPosition="86" Width="40" Height="14"
Name="TANK_SIZE" ItemType="Text Item" CanvasName="CANVAS" Prompt="Velikost
nádrže" PromptAttachmentOffset="5" PromptAttachmentEdge="Left"/>
        <Item XPosition="268" YPosition="86" Width="40" Height="14"
Name="GAS_MILEAGE" ItemType="Text Item" CanvasName="CANVAS" Prompt="Průměrná
spotřeba" PromptAttachmentOffset="5" PromptAttachmentEdge="Left"/>
        <Item XPosition="108" YPosition="101" Width="200" Height="14"
Name="ID_USER" ItemType="Text Item" PromptAttachmentOffset="5"/>

```



```

        <Item xmlns="" YPosition="104" XPosition="110"
PromptAttachmentEdge="Left" Width="195" Height="14" Name="USER_NAME"
ItemType="Text Item" Prompt="Zodpovědný<br>pracovník"
CanvasName="CANVAS" PromptAttachmentOffset="5" LovName="LOV_ID_USER"
DatabaseItem="false" ValidateFromList="true"/>
    </Block>
    <RecordGroup Name="RG_ID_USER" RecordGroupQuery="select NAME, ID from
QAX_USERS">
        <RecordGroupColumn Name="NAME"/>
        <RecordGroupColumn Name="ID"/>
    </RecordGroup>
    <LOV Name="LOV_ID_USER" RecordGroupName="RG_ID_USER">
        <LOVColumnMapping Name="NAME" DisplayWidth="100" Title="Jméno a
příjmení" ReturnItem="VX_CL_CARS.USER_NAME"/>
        <LOVColumnMapping Name="ID" Title="ID"
ReturnItem="VX_CL_CARS.ID_USER"/>
    </LOV>
</FormModule>
</Module>

<!-- F_P_VX_CL_LOG -->
<Module xmlns="http://xmlns.oracle.com/Forms">
    <FormModule ConsoleWindow="WINDOW" Name="F_P_VX_CL_LOG">
        <Block Name="CONTROL">
            <Item YPosition="16" XPosition="48" Width="175" Height="19"
Name="CAR" ItemType="List Item" CanvasName="CANVAS" Prompt="Vozidlo"
PromptAttachmentOffset="5" PostChange="true"/>
            <Item Name="DATE_FROM" CanvasName="CANVAS" PostChange="true"
ItemType="Text Item" XPosition="270" YPosition="16" Width="58" Height="16"
Prompt="Rok" PromptAttachmentOffset="5"/>
            <Item YPosition="16" XPosition="379" PromptAttachmentEdge="Left"
Width="58" Height="19" Name="DATE_TO" Prompt="Měsíc" CanvasName="CANVAS"
PromptAttachmentOffset="5" PostChange="true" ItemType="List Item" Value="00">
                <ListItemElement xmlns="" Name="-" Value="00"/>
                <ListItemElement xmlns="" Name="1" Value="01"/>
                <ListItemElement xmlns="" Name="2" Value="02"/>
                <ListItemElement xmlns="" Name="3" Value="03"/>
                <ListItemElement xmlns="" Name="4" Value="04"/>
                <ListItemElement xmlns="" Name="5" Value="05"/>
                <ListItemElement xmlns="" Name="6" Value="06"/>
                <ListItemElement xmlns="" Name="7" Value="07"/>
                <ListItemElement xmlns="" Name="8" Value="08"/>
                <ListItemElement xmlns="" Name="9" Value="09"/>
                <ListItemElement xmlns="" Name="10" Value="10"/>
                <ListItemElement xmlns="" Name="11" Value="11"/>
                <ListItemElement xmlns="" Name="12" Value="12"/>
            </Item>
            <Item YPosition="312" XPosition="372" PromptAttachmentEdge="Left"
Width="51" Height="14" Name="SUM" ItemType="Text Item" Prompt="Suma km"
CanvasName="CANVAS" PromptAttachmentOffset="5" Enabled="false"
Visible="false"/>
        </Block>
        <Canvas Name="CANVAS" WindowName="WINDOW" Width="620" Height="332"
ViewportXPosition="1" ViewportYPosition="44"/>
        <Canvas Name="SYSCANVAS_TOP" Width="620" Height="40"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="1" ViewportWidth="undefined" ViewportHeight="undefined">
        </Canvas>
    </FormModule>
</Module>

```

```

    <Canvas Name="SYSCANVAS_LOW" Width="620" Height="18"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="379" ViewportWidth="undefined" ViewportHeight="undefined">
    </Canvas>
    <Block RecordsDisplayCount="10" Name="VX_CL_LOG" Width="591"
QueryDataSourceName="QVX_CL_LOG" DatabaseBlock="true" InsertAllowed="false"
UpdateAllowed="false" Height="330" ShowFilters="true"
OrderBy="to_date(log_date) desc, time ASC" AllowInsertTemp="false">
        <Item Width="50" Height="16" Name="DETAIL" ItemType="Push Button"
CanvasName="CANVAS" Label="    Detail" StyleClass="key_edit" XPosition="17"
YPosition="42"/>
        <Item Width="60" Height="18" Name="LOG_DATE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Datum" Enabled="false"/>
        <Item Width="85" Height="18" Name="TIME" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Čas (start - cíl)" Enabled="false"/>
        <Item Width="140" Height="18" Name="LOG_NOTE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Popis" Enabled="false"/>
        <Item Width="60" Height="18" Name="DISTANCE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Vzdálenost" Sum="true" Enabled="false"
DataType="NUMBER"/>
        <Item Width="80" Height="18" Name="B_P_TXT" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Typ jízdy" EnumFilter="true" Enabled="false"/>
        <Item Width="70" Height="18" Name="LOG_USE_TXT" ItemType="Text
Item" CanvasName="CANVAS" Prompt="Účel jízdy" Enabled="false"/>
        <Item Width="100" Height="18" Name="ID_PROJECTS" ItemType="Text
Item" CanvasName="CANVAS" Prompt="Projekt" Visible="false" Enabled="false"/>
        <Item Width="50" Height="18" Name="STATUS" ItemType="List Item"
CanvasName="CANVAS" Prompt="Stav" Visible="false" Enabled="false">
            <ListItemElement Name="Rozpracováno" Value="RO"/>
            <ListItemElement Name="Schváleno" Value="SM"/>
        </Item>
        <Item Width="100" Height="18" Name="ID_DOCUMENTS" ItemType="Text
Item" CanvasName="CANVAS" Prompt="Úkol" Visible="false" Enabled="false"/>
        <Item Width="100" Height="18" Name="ID" ItemType="Text Item"
CanvasName="CANVAS" Visible="false"/>
        <Item Width="100" Height="18" Name="ID_CAR" ItemType="Text Item"
CanvasName="CANVAS" Visible="false"/>
    </Block>
    <Block Name="SYSBLOCK">
        <Item YPosition="3" XPosition="583" Width="30" Height="18"
Name="VERZE" ItemType="HTML" CanvasName="SYSCANVAS_TOP" Value="v. 1.01"/>
        <Item YPosition="3" XPosition="5" Width="416" Height="18"
Name="TOPIC" ItemType="HTML" CanvasName="SYSCANVAS_TOP"/>
        <Item YPosition="19" XPosition="5" Width="20" Height="20"
Name="KEY_NEW" Enabled="true" ItemType="Push Button" CanvasName="SYSCANVAS_TOP"
StyleClass="key_new" Tooltip="Nový záznam"/>
        <Item YPosition="19" XPosition="25" Width="20" Height="20"
Name="KEY_EDIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_edit" Tooltip="Upravit záznam"/>
        <Item YPosition="19" XPosition="45" Width="20" Height="20"
Name="KEY_DELETE" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_delete" Tooltip="Smazat záznam"/>
        <Item YPosition="19" XPosition="70" Width="20" Height="20"
Name="KEY_COMMIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_commit" Tooltip="Uložit"/>
        <Item YPosition="19" XPosition="90" Width="20" Height="20"
Name="KEY_REFRESH" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_refresh" Tooltip="Obnovit data"/>

```

```

        <Item YPosition="19" XPosition="115" Width="20" Height="20"
Name="KEY_HELP" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_help" Tooltip="Nápověda"/>
        <Item YPosition="19" XPosition="135" Width="20" Height="20"
Name="KEY_HISTORY" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_history" Tooltip="Historie
záznamu"/>
        <Item YPosition="19" XPosition="155" Width="20" Height="20"
Name="KEY_PRINT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_print" Tooltip="Tisk"/>
        <Item YPosition="19" XPosition="175" Width="20" Height="20"
Name="KEY_XLSX" Enabled="true" ItemType="File Download"
CanvasName="SYSCANVAS_TOP" StyleClass="key_xlsx" Tooltip="Export do excelu"
ItemProcedure="KEY_XLSX" ItemProcedureParam="VX_CL_LOG"/>
        <Item YPosition="19" XPosition="200" Width="20" Height="20"
Name="KEY_SETTINGS" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_settings" Tooltip="Nastavení"
ItemProcedureParam="VX_CL_LOG" ItemProcedure="KEY_SETTINGS"/>
        <Item YPosition="1" XPosition="1" Width="457" Height="18"
Name="STATUSBAR" ItemType="HTML" CanvasName="SYSCANVAS_LOW"/>
    </Block>
</FormModule>
</Module>

<!-- F_E_VX_CL_LOG -->
<Module xmlns="http://xmlns.oracle.com/Forms">
    <FormModule ConsoleWindow="WINDOW" Name="F_E_VX_CL_LOG">
        <Block Name="CONTROL">
            <Item YPosition="8" XPosition="18" Width="100" Height="18"
Name="POP_CL_LOG" ItemType="HTML" CanvasName="CANVAS" Value="Detail jízdy"/>
            <Item YPosition="35" XPosition="437" Width="50" Height="17"
Name="BUTT_NEW" ItemType="Push Button" CanvasName="CAN_TAB" Label="Nový"
StyleClass="key_new"/>
            <Item xmlns="" YPosition="48" XPosition="442" Width="36"
Height="14" Name="PREVIOUS" ItemType="Text Item" CanvasName="CANVAS"
LovName="LOV_PREVIOUS" Prompt="Předchozí jízdy" PromptAttachmentOffset="5"/>
            <Item xmlns="" YPosition="79" XPosition="473" Width="34"
Height="14" Name="BUTT_DOPOCTI" ItemType="Push Button" CanvasName="CANVAS"
Label="Dopočti"/>
        </Block>
        <Canvas Name="CANVAS" WindowName="WINDOW" Width="530" Height="448"
ViewportXPosition="1" ViewportYPosition="43">
            <Graphics Name="RAM" XPosition="19" YPosition="24" Width="492"
Height="112"/>
        </Canvas>
        <Canvas Name="SYSCANVAS_TOP" Width="530" Height="40"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="0" ViewportWidth="undefined" ViewportHeight="undefined">
        </Canvas>
        <Canvas Name="SYSCANVAS_LOW" Width="530" Height="18"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="494" ViewportWidth="undefined" ViewportHeight="undefined">
        </Canvas>
        <Block Name="VX_CL_LOG" QueryDataSourceName="VX_CL_LOG"
WhereClause="ID=:VX_CL_LOG.ID">
            <Item YPosition="2" XPosition="333" Width="21" Height="13"
Name="ID" ItemType="Text Item" CanvasName="CANVAS" Prompt="ID" Enabled="false"
Visible="false"/>

```

```

        <Item YPosition="2" XPosition="393" Width="26" Height="14"
Name="ID_CAR" ItemType="Text Item" CanvasName="CANVAS" Prompt="ID_CAR"
Enabled="false" Visible="false"/>
        <Item YPosition="63" XPosition="86" Width="110" Height="14"
Name="ID_PROJECTS" ItemType="Text Item" CanvasName="CANVAS" Prompt="Projekt"
LovName="LOV_PROJECTS" PromptAttachmentOffset="5"/>
        <Item Name="LOG_USE" ItemType="List Item" CanvasName="CANVAS"
XPosition="244" YPosition="63" Width="93" Height="17" Prompt="Účel jízdy"
PromptAttachmentOffset="5"/>
        <Item Width="90" Height="17" Name="B_P" ItemType="List Item"
CanvasName="CANVAS" Prompt="Typ jízdy" XPosition="379" YPosition="63"
PromptAttachmentOffset="5">
            <ListItemElement Name="Služební" Value="B"/>
            <ListItemElement Name="Soukromé" Value="P"/>
        </Item>
        <Item YPosition="79" XPosition="86" Width="50" Height="14"
Name="LOG_DATE" ItemType="Text Item" CanvasName="CANVAS" Prompt="Datum"
DataType="DATE" PromptAttachmentOffset="5" FormatMask="DD.MM.YYYY"/>
        <Item YPosition="79" XPosition="205" Width="50" Height="14"
Name="START_TIME" ItemType="Text Item" CanvasName="CANVAS" Prompt="Čas startu"
PromptAttachmentOffset="5"/>
        <Item YPosition="79" XPosition="302" Width="50" Height="14"
Name="END_TIME" ItemType="Text Item" CanvasName="CANVAS" Prompt="Čas cíle"
PromptAttachmentOffset="5"/>
        <Item YPosition="79" XPosition="416" Width="50" Height="14"
Name="DISTANCE" ItemType="Text Item" CanvasName="CANVAS" Prompt="Vzdálenost"
PromptAttachmentOffset="5" DataType="NUMBER"/>
        <Item YPosition="94" XPosition="86" Width="380" Height="14"
Name="LOG_NOTE" ItemType="Text Item" CanvasName="CANVAS" Prompt="Popis"
PromptAttachmentOffset="5"/>
        <Item Width="100" Height="16" Name="STATUS" ItemType="List Item"
CanvasName="CANVAS" Prompt="Stav" XPosition="86" YPosition="109"
PromptAttachmentOffset="5" Visible="false">
            <ListItemElement Name="Rozpracováno" Value="RO"/>
            <ListItemElement Name="Schváleno" Value="SM"/>
        </Item>
        <Item YPosition="109" XPosition="222" Width="100" Height="14"
Name="ID_DOCUMENTS" ItemType="Text Item" CanvasName="CANVAS" Prompt="Úkol"
Visible="false" PromptAttachmentOffset="5"/>
    </Block>
    <RecordGroup Name="RG_PROJECTS" RecordGroupQuery="select LONG_NAME,
short_name from QVX_PROJECTS">
        <RecordGroupColumn Name="LONG_NAME"/>
        <RecordGroupColumn xmlns="" Name="SHORT_NAME"/>
    </RecordGroup>
    <LOV Name="LOV_PROJECTS" RecordGroupName="RG_PROJECTS">
        <LOVColumnMapping xmlns="" Name="SHORT_NAME" Title="Zkratka"
DisplayWidth="80" ReturnItem="VX_CL_LOG.ID_PROJECTS"/>
        <LOVColumnMapping Name="LONG_NAME" Title="Název"
DisplayWidth="100"/>
    </LOV>
    <Block Name="SYSBLOCK">
        <Item YPosition="3" XPosition="496" Width="30" Height="18"
Name="VERZE" ItemType="HTML" CanvasName="SYSCANVAS_TOP" Value="v. 1.01"/>
        <Item YPosition="3" XPosition="5" Width="416" Height="18"
Name="TOPIC" ItemType="HTML" CanvasName="SYSCANVAS_TOP"/>

```

```

        <Item YPosition="19" XPosition="5" Width="20" Height="20"
Name="KEY_NEW" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_new" Tooltip="Nový záznam"/>
        <Item YPosition="19" XPosition="25" Width="20" Height="20"
Name="KEY_EDIT" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_edit" Tooltip="Upravit záznam"/>
        <Item YPosition="19" XPosition="45" Width="20" Height="20"
Name="KEY_DELETE" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_delete" Tooltip="Smazat záznam"/>
        <Item YPosition="19" XPosition="70" Width="20" Height="20"
Name="KEY_COMMIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_commit" Tooltip="Uložit"/>
        <Item YPosition="19" XPosition="90" Width="20" Height="20"
Name="KEY_REFRESH" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_refresh" Tooltip="Obnovit data"/>
        <Item YPosition="19" XPosition="115" Width="20" Height="20"
Name="KEY_HELP" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_help" Tooltip="Nápověda"/>
        <Item YPosition="19" XPosition="135" Width="20" Height="20"
Name="KEY_HISTORY" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_history" Tooltip="Historie
záznamu"/>
        <Item YPosition="19" XPosition="155" Width="20" Height="20"
Name="KEY_PRINT" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_print" Tooltip="Tisk"/>
        <Item YPosition="19" XPosition="175" Width="20" Height="20"
Name="KEY_XLSX" Enabled="false" ItemType="File Download"
CanvasName="SYSCANVAS_TOP" StyleClass="key_xlsx" Tooltip="Export do excelu"
ItemProcedure="KEY_XLSX" ItemProcedureParam="QAX_USERS"/>
        <Item YPosition="19" XPosition="200" Width="20" Height="20"
Name="KEY_SETTINGS" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_settings" Tooltip="Nastavení"
ItemProcedure="KEY_SETTINGS"/>
        <Item YPosition="1" XPosition="1" Width="457" Height="18"
Name="STATUSBAR" ItemType="HTML" CanvasName="SYSCANVAS_LOW"/>
    </Block>
    <Block RecordsDisplayCount="10" Name="VX_CL_FUEL_LOG" Height="274"
Width="425" QueryDataSourceName="VX_CL_FUEL_LOG" DatabaseBlock="true"
InsertAllowed="false" UpdateAllowed="true" WhereClause="ID_LOG =:VX_CL_LOG.ID"
AllowInsertTemp="false">
        <Item Width="30" Height="18" Name="ID" ItemType="Text Item"
CanvasName="CAN_TAB" XPosition="6" YPosition="32" Visible="false" Prompt="ID"/>
        <Item Width="100" Height="18" Name="ID_CAR" ItemType="Text Item"
CanvasName="CAN_TAB" Visible="false"/>
        <Item Width="50" Height="18" Name="ID_LOG" ItemType="Text Item"
CanvasName="CAN_TAB" Visible="false" Prompt="ID_LOG"/>
        <Item Width="60" Height="18" Name="AMOUT" ItemType="Text Item"
CanvasName="CAN_TAB" Prompt="Množství<br> paliva" Sum="true"
DataType="NUMBER" FormatMask="999D99"/>
        <Item Width="60" Height="18" Name="PRICE" ItemType="Text Item"
CanvasName="CAN_TAB" Prompt="Cena" Sum="true" DataType="NUMBER"
FormatMask="999G999D99"/>
        <Item Width="70" Height="18" Name="MILEAGE" ItemType="Text Item"
CanvasName="CAN_TAB" Prompt="Stav<br> tachometru" DataType="NUMBER"/>
        <Item Width="140" Height="18" Name="LOG_NOTE" ItemType="Text Item"
CanvasName="CAN_TAB" Prompt="Poznámka"/>
        <Item Width="50" Height="18" Name="BUTT_DEL" ItemType="Push Button"
CanvasName="CAN_TAB" StyleClass="key_delete" Label="    Smaž"/>

```



```

        </Block>
        <Canvas WindowName="WINDOW" Name="CAN_TAB" ViewportXPosition="21"
ViewportYPosition="191" Width="491" Height="285">
            <TabPage Name="CL_FUEL_LOG_ID" Label="Čerpání"/>
        </Canvas>
        <Block Name="VX_CL_CARS" DatabaseBlock="true" InsertAllowed="false"
UpdateAllowed="true" QueryDataSourceName="Q_VX_CL_CARS" WhereClause="ID
=:VX_CL_LOG.ID_CAR">
            <Item YPosition="47" XPosition="244" Width="90" Height="14"
Name="REG_PLATE" ItemType="Text Item" CanvasName="CANVAS" Prompt="SPZ"
Enabled="false" PromptAttachmentEdge="Top" PromptAttachmentOffset="5"/>
            <Item YPosition="47" XPosition="86" Width="110" Height="14"
Name="USER_NAME" ItemType="Text Item" CanvasName="CANVAS" Prompt="Zodpovědný
pracovník" Enabled="false" PromptAttachmentEdge="Top"
PromptAttachmentOffset="5"/>
            <Item YPosition="1" XPosition="486" Name="ID" ItemType="Text Item"
CanvasName="CANVAS" Prompt="ID (CL_CARS)" Enabled="false" Visible="false"
Width="23" Height="14"/>
        </Block>
        <RecordGroup Name="RG_PREVIOUS" RecordGroupQuery="select start_time,
end_time, log_note, distance, decode(B_P, 'B', 'Služební', 'P', 'Pracovní') AS
B_P_TXT, decode(LOG_USE, 'PRAC', 'Pracovní', 'JINY', 'Jiný', 'OBCHOD',
'Obchodní', 'SERVIS', 'Servis vozu') AS LOG_USE_TXT, max(log_date) as datum
from VX_CL_LOG where ID_CAR=:VX_CL_LOG.ID_CAR group by start_time, end_time,
log_note, distance, b_p, log_use order by datum desc">
            <RecordGroupColumn Name="LOG_NOTE"/>
            <RecordGroupColumn Name="START_TIME"/>
            <RecordGroupColumn Name="END_TIME"/>
            <RecordGroupColumn Name="DISTANCE"/>
            <RecordGroupColumn Name="B_P_TXT"/>
            <RecordGroupColumn Name="LOG_USE_TXT"/>
            <RecordGroupColumn Name="DATUM"/>
        </RecordGroup>
        <LOV Name="LOV_PREVIOUS" RecordGroupName="RG_PREVIOUS">
            <LOVColumnMapping Name="DATUM" DisplayWidth="75" Title="Datum"/>
            <LOVColumnMapping Name="START_TIME"
ReturnItem="VX_CL_LOG.START_TIME" Title="Čas<br>startu"
DisplayWidth="50"/>
            <LOVColumnMapping Name="END_TIME" Title="Čas<br>cíle"
ReturnItem="VX_CL_LOG.END_TIME" DisplayWidth="50"/>
            <LOVColumnMapping Name="LOG_NOTE" DisplayWidth="220" Title="Popis"
ReturnItem="VX_CL_LOG.LOG_NOTE"/>
            <LOVColumnMapping Name="DISTANCE" ReturnItem="VX_CL_LOG.DISTANCE"
DisplayWidth="85" Title="Vzdálenost"/>
            <LOVColumnMapping Name="B_P_TXT" DisplayWidth="60" Title="Typ
<br>jízdy" ReturnItem="VX_CL_LOG.B_P"/>
            <LOVColumnMapping Name="LOG_USE_TXT" DisplayWidth="60" Title="Účel
jízdy" ReturnItem="VX_CL_LOG.LOG_USE"/>
        </LOV>
    </FormModule>
</Module>

<!-- F_P_VX_LOG_OVERVIEW -->
<Module xmlns="http://xmlns.oracle.com/Forms">
    <FormModule ConsoleWindow="WINDOW" Name="F_P_VX_LOG_OVERVIEW">
        <Block Name="CONTROL">

```

```

        <Item YPosition="16" XPosition="48" Width="175" Height="19"
Name="CAR" ItemType="List Item" CanvasName="CANVAS" Prompt="Vozidlo"
PromptAttachmentOffset="5" PostChange="true" Enabled="true" Value="00">
        <ListItemElement xmlns="" Name="-" Value="00"/>
    </Item>
    <Item Name="DATE_FROM" CanvasName="CANVAS" PostChange="true"
ItemType="Text Item" XPosition="270" YPosition="16" Width="58" Height="16"
Prompt="Rok" PromptAttachmentOffset="5" Enabled="true"/>
    <Item YPosition="16" XPosition="379" PromptAttachmentEdge="Left"
Width="58" Height="19" Name="DATE_TO" Prompt="Měsíc" CanvasName="CANVAS"
PromptAttachmentOffset="5" ItemType="List Item" Value="00" Enabled="true"
PostChange="true">
        <ListItemElement xmlns="" Name="-" Value="00"/>
        <ListItemElement xmlns="" Name="1" Value="01"/>
        <ListItemElement xmlns="" Name="2" Value="02"/>
        <ListItemElement xmlns="" Name="3" Value="03"/>
        <ListItemElement xmlns="" Name="4" Value="04"/>
        <ListItemElement xmlns="" Name="5" Value="05"/>
        <ListItemElement xmlns="" Name="6" Value="06"/>
        <ListItemElement xmlns="" Name="7" Value="07"/>
        <ListItemElement xmlns="" Name="8" Value="08"/>
        <ListItemElement xmlns="" Name="9" Value="09"/>
        <ListItemElement xmlns="" Name="10" Value="10"/>
        <ListItemElement xmlns="" Name="11" Value="11"/>
        <ListItemElement xmlns="" Name="12" Value="12"/>
    </Item>
</Block>
<Canvas Name="CANVAS" WindowName="WINDOW" Width="862" Height="417"
ViewportXPosition="1" ViewportYPosition="44"/>
    <Block RecordsDisplayCount="10" Name="VX_CL_LOG_VIEW" Height="429"
Width="852" QueryDataSourceName="QVX_CL_LOG_OVERVIEW" ShowFilters="true"
DatabaseBlock="true" InsertAllowed="false" UpdateAllowed="false"
OrderBy="to_date(log_date) desc, a_time asc">
        <Item Width="100" Height="18" Name="ID" ItemType="Text Item"
CanvasName="CANVAS" XPosition="1" YPosition="48" Visible="false"/>
        <Item Width="60" Height="18" Name="LOG_DATE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Datum"/>
        <Item Width="60" Height="18" Name="A_TIME" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Čas<br>(start - cíl)/>
        <Item Width="130" Height="18" Name="LOG_NOTE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Popis"/>
        <Item Width="60" Height="18" Name="DISTANCE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Vzdálenost" DataType="NUMBER"/>
        <Item Width="70" Height="18" Name="B_P_TXT" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Typ jízdy" EnumFilter="true" Enabled="false"/>
        <Item Width="75" Height="18" Name="LOG_USE_TXT" ItemType="Text
Item" CanvasName="CANVAS" Prompt="Účel jízdy" Enabled="false"/>
        <Item Width="100" Height="18" Name="USER_NAME" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Vlastník"/>
        <Item Width="60" Height="18" Name="REG_PLATE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="SPZ"/>
        <Item xmlns="" Width="90" Height="18" Name="LONG_NAME"
ItemType="Text Item" Prompt="Vozidlo" CanvasName="CANVAS"/>
        <Item Width="40" Height="18" Name="S_PRICE" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Cena" DataType="NUMBER" FormatMask="999G990D99"/>
        <Item Width="50" Height="18" Name="S_AMOUT" ItemType="Text Item"
CanvasName="CANVAS" Prompt="Množství<br>paliva" DataType="NUMBER"
FormatMask="999G990D99"/>

```

```

        <Item xmlns="" Width="100" Height="18" Name="ID_CAR" ItemType="Text
Item" CanvasName="CANVAS" Visible="false"/>
    </Block>
    <Canvas Name="SYSCANVAS_TOP" Width="862" Height="40"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="1" ViewportWidth="undefined" ViewportHeight="undefined">
    </Canvas>
    <Canvas Name="SYSCANVAS_LOW" Width="862" Height="18"
WindowName="WINDOW" CanvasType="undefined" ViewportXPosition="1"
ViewportYPosition="464" ViewportWidth="undefined" ViewportHeight="undefined">
    </Canvas>
    <Block Name="SYSBLOCK">
        <Item YPosition="6" XPosition="830" Width="30" Height="18"
Name="VERZE" ItemType="HTML" CanvasName="SYSCANVAS_TOP" Value="v. 1.01"/>
        <Item YPosition="3" XPosition="5" Width="416" Height="18"
Name="TOPIC" ItemType="HTML" CanvasName="SYSCANVAS_TOP"/>
        <Item YPosition="19" XPosition="5" Width="20" Height="20"
Name="KEY_NEW" Enabled="true" ItemType="Push Button" CanvasName="SYSCANVAS_TOP"
StyleClass="key_new" Tooltip="Nový záznam"/>
        <Item YPosition="19" XPosition="25" Width="20" Height="20"
Name="KEY_EDIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_edit" Tooltip="Upravit záznam"/>
        <Item YPosition="19" XPosition="45" Width="20" Height="20"
Name="KEY_DELETE" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_delete" Tooltip="Smazat záznam"/>
        <Item YPosition="19" XPosition="70" Width="20" Height="20"
Name="KEY_COMMIT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_commit" Tooltip="Uložit"/>
        <Item YPosition="19" XPosition="90" Width="20" Height="20"
Name="KEY_REFRESH" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_refresh" Tooltip="Obnovit data"/>
        <Item YPosition="19" XPosition="115" Width="20" Height="20"
Name="KEY_HELP" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_help" Tooltip="Nápověda"/>
        <Item YPosition="19" XPosition="135" Width="20" Height="20"
Name="KEY_HISTORY" Enabled="false" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_history" Tooltip="Historie
záznamu"/>
        <Item YPosition="19" XPosition="155" Width="20" Height="20"
Name="KEY_PRINT" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_print" Tooltip="Tisk"/>
        <Item YPosition="19" XPosition="175" Width="20" Height="20"
Name="KEY_XLSX" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_xlsx" Tooltip="Export do excelu"
ItemProcedure="KEY_XLSX" ItemProcedureParam="VX_CL_LOG_VIEW"/>
        <Item YPosition="19" XPosition="200" Width="20" Height="20"
Name="KEY_SETTINGS" Enabled="true" ItemType="Push Button"
CanvasName="SYSCANVAS_TOP" StyleClass="key_settings" Tooltip="Nastavení"
ItemProcedureParam="VX_CL_LOG" ItemProcedure="KEY_SETTINGS"/>
        <Item YPosition="1" XPosition="1" Width="457" Height="18"
Name="STATUSBAR" ItemType="HTML" CanvasName="SYSCANVAS_LOW"/>
    </Block>
</FormModule>
</Module>

```



## (B) Kód packages

```
create or replace PACKAGE BODY      "F_P_VX_CL_CARS" AS

PRAGMA SERIALLY_REUSABLE;

procedure SETFORM ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS );
-----
-
-- WHEN NEW FORMS INSTANCE
-----
-
procedure WHEN_NEW_FORMS_INSTANCE (ParAXForm in out nocopy
PAX_FORMS.TYPE_ITEMS) AS
cnt number;
BEGIN
  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE started. ');
  PAX_FORMS_UTL.HIDETOOLBAR (ParAXForm);
  ParAXForm('TOOLBAR.KEY_COMMIT').ENABLED:='TRUE';
  SETFORM(paraxform);
  select count(*) into cnt from VX_CL_CARS;
  paraxform('SYSBLOCK.TOPIC').val := '<b>' || PAX_LANG.getmsg('Přehled
vozidel', 'CZ') || '</b>' ( ' || PAX_LANG.getmsg('Počet záznamů', 'CZ') || ':
' || cnt || ' ');
  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE ended. ');

END WHEN_NEW_FORMS_INSTANCE;

-----
-
-- TOOLBAR
-----
-
procedure KEY_NEW (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS)AS
newCompId varchar2(256);
editFormName varchar2(256) := 'F_E_VX_CL_CARS';
parReadOnly varchar2(5) := 'false';

BEGIN
  newCompId:=PAX_FORMS_WIN.CREATEFORMWINDOW(editFormName,PAX_LANG.getmsg('Nový
záznam', 'CZ'), 'true');
  PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_ID', null);
  PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_READ_ONLY', parReadOnly);
END KEY_NEW;

procedure KEY_EDIT (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
prow rowid;
BEGIN
  select rowid into prow from VX_CL_CARS where
ROWIDTOCHAR(rowid)=paraxform('VX_CL_CARS.AXCURRENTROW').val;
  detail(paraxform,prow);
END KEY_EDIT;

procedure KEY_HELP (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
```

```

    v_help_file varchar2(600) := '/Dokumentace/Administrace systému/Evidence
vozů.pdf';
    file_id number;
BEGIN
    select id into file_id from vx_dirdata where pvx_files.get_full_name(id) =
v_help_file;
    PAX_FORMS_UTL.openURL(ParAXForm, 'filevx/' || file_id, false);
END KEY_HELP;

procedure KEY_SETTINGS ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS,pBlock in
varchar2) as
begin
    PAX_AXTABLE.showColumnEditor(PAX_AXTABLE.GetComponentID(ParAXForm,pBlock));
exception when others then
    AXSTACKTRACE;
end;

procedure KEY_XLSX ( parAXForm IN OUT NOCOPY AXFORMS.TYPE_ITEMS, pBlock in
varchar2, parBlobData OUT BLOB, parFileContentType OUT VARCHAR2, parFileName
OUT VARCHAR2 ) as
    vQuery VARCHAR2(31024);
    vDatasource varchar2(255) := pBlock;
    idc varchar2(32000) := PAX_AXTABLE.GetComponentID(ParAXForm, vDatasource);
    visibleColumns VARCHAR2(32767);
begin
    visibleColumns := PAX_AXTABLE.getVisibleColumnsList(idc);
    visibleColumns := replace (visibleColumns, 'AX__ROWID,', '');
    visibleColumns := replace (visibleColumns, 'DETAIL,', '');
    vQuery := 'SELECT ' || visibleColumns || ' FROM ( ' || PAX_PAGE.getServerPar
(idc,'dataQueryNoFormat') || ')';
    --vQuery := pax_page.getserverpar (PAX_AXTABLE.GetComponentID(ParAXForm,
'QAKCE'),'dataQueryNoFormat');
    axLogDebug('sqlSource: ' || vQuery);
    DBMS_LOB.CreateTemporary(parBlobData, TRUE);
    axlogdebug('***** GENEXCEL START
*****');
    parblobData := PAX_OFFICE.gen_excel(vQuery,vDatasource);
    axlogdebug('***** GENEXCEL END
*****');
    parFileContentType:='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet';
    parFileName:=vDatasource||'.xlsx';
exception when others then
    AXSTACKTRACE;
end;
-----
-
-- FORM METHODS
-----
-
procedure DETAIL (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS, prow rowid,
pReadOnly in varchar2 default 'false') as
    newCompId varchar2(256);
    lockTable varchar2(256) := 'VX_CL_CARS';
    editFormName varchar2(256) := 'F_E_VX_CL_CARS';
    pId number;
    vChyba varchar2(20000);
    parReadOnly varchar2(5) := pReadOnly;

```

```

begin
  axlogdebug('Procedure DETAIL started.');
```

```

  select id into pid from VX_CL_CARS where rowid = prow;
  if pid is null then
    pax_forms.dialogok(paraxform,PAX_LANG.getmsg('Vyberte
záznam', 'CZ'),PAX_LANG.getmsg('Chyba', 'CZ'),2);
    return;
  end if;

  --zamknu si objekt
  if parReadOnly = 'false' then
    if not PAX_FORMS_WIN.Lockobject (lockTable,pId,vChyba) then
      parReadOnly := 'true';
      pax_forms.dialogok(paraxform,vChyba,PAX_LANG.getmsg('Informace', 'CZ'));
    end if;
  end if;

  --založím okno

newCompId:=PAX_FORMS_WIN.CREATEFORMWINDOW(editFormName,PAX_LANG.getmsg('Detail
vozidla', 'CZ'),'true');
  PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_ID',pid);
  PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_READ_ONLY',parReadOnly);
  axlogdebug('Procedure DETAIL ended.');
```

```

end DETAIL;
```

```

procedure SETFORM ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS ) as
  mTableID varchar2(256);
begin
  --veci co maji byt pristupne vzdy dostanou forceenable true
  --veci co maji byt pristupne pouze pokud ma uzivatel pristup na zapis
dostanou enabled true
  --veci co nemaji byt pristupne dostanou enabled false
  --VNITRNI TOOLBAR
  paraxform('SYSBLOCK.KEY_NEW').enabled:='true';
  paraxform('SYSBLOCK.KEY_EDIT').enabled:='true';
  paraxform('SYSBLOCK.KEY_EDIT').forceenable:='true';
  paraxform('SYSBLOCK.KEY_DELETE').enabled:='false';
  paraxform('SYSBLOCK.KEY_COMMIT').enabled:='false';
  paraxform('SYSBLOCK.KEY_HELP').enabled:='true';
  paraxform('SYSBLOCK.KEY_HELP').forceenable:='true';
  paraxform('SYSBLOCK.KEY_HISTORY').enabled:='true';
  paraxform('SYSBLOCK.KEY_HISTORY').forceenable:='true';
  paraxform('SYSBLOCK.KEY_PRINT').enabled:='false';
  paraxform('SYSBLOCK.KEY_SETTINGS').enabled:='true';
  paraxform('SYSBLOCK.KEY_SETTINGS').forceenable:='true';
  paraxform('SYSBLOCK.KEY_XLSX').enabled:='true';
  paraxform('SYSBLOCK.KEY_XLSX').forceenable:='true';
end;
```

```

END "F_P_VX_CL_CARS";

create or replace PACKAGE BODY          "F_E_VX_CL_CARS" AS

PRAGMA SERIALLY_REUSABLE;
```

```
-----  
-  
-- WHEN NEW FORMS INSTANCE  
-----  
-
```

```
procedure WHEN_NEW_FORMS_INSTANCE (ParAXForm in out nocopy  
PAX_FORMS.TYPE_ITEMS) AS  
BEGIN  
  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE started.');
```

```
  PAX_FORMS_UTL.HIDETOOLBAR (ParAXForm);  
  if GETPAR(paraxform,'par_read_only') = 'false' then  
    ParAXForm('TOOLBAR.KEY_COMMIT').ENABLED:='TRUE';  
  end if;  
  paraxform('VX_CL_CARS.ID').val := GETPAR(paraxform,'par_id');  
  PAX_FORMS.AXREFRESH(paraxform);  
  if paraxform('VX_CL_CARS.ID').val is not null then  
    begin  
      select first_name||' '||last_name into paraxform('VX_CL_CARS.USER_NAME').val  
from ax_users where id=paraxform('VX_CL_CARS.ID_USER').val;  
      exception when others then  
        axlog('Chyba načtení uživatele');  
      end;  
    end if;  
  -- PAX_FORMS_UTL.setBlockReset(ParAXForm, 'VX_CL_CARS');  
  -- AX_FORMS_UTL.reinitFormTable(ParAXForm, 'VX_CL_CARS');  
  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE ended.');
```

```
END WHEN_NEW_FORMS_INSTANCE;
```

```
-----  
-  
-- TOOLBAR  
-----  
-
```

```
procedure KEY_COMMIT (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS  
BEGIN  
  if paraxform('VX_CL_CARS.ID').val is null then  
    paraxform('VX_CL_CARS.ID').val := pax_forms.axinsert (paraxform,  
'VX_CL_CARS.ID');  
    commit;
```

```
  pax_page.setserverpar(PAX_FORMS_UTL.GETCOMPONENTID(paraxform),'par_id',paraxfor  
m('VX_CL_CARS.ID').val);  
  else  
    pax_forms.axupdate(paraxform);  
    commit;  
  end if;  
  PAX_FORMS.SETINACTIVE(paraxform);  
END KEY_COMMIT;
```

```
procedure KEY_REFRESH (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS  
BEGIN  
  rollback;  
  PAX_PAGE.SETCOMPONENTDIRTY(PAX_FORMS_UTL.getComponentId(PARAXFORM));  
END KEY_REFRESH;
```

```
-----  
-  
-- FORM METHODS  
-----  
-
```

```
END "F_E_VX_CL_CARS";
```

```
create or replace PACKAGE BODY "F_P_VX_CL_LOG" AS
```

```
PRAGMA SERIALLY_REUSABLE;
```

```
procedure SETFORM ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS );
```

```
procedure load_date (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS);  
-----  
-
```

```
-- WHEN NEW FORMS INSTANCE  
-----  
-
```

```
procedure WHEN_NEW_FORMS_INSTANCE (ParAXForm in out nocopy  
PAX_FORMS.TYPE_ITEMS) AS
```

```
cnt number;
```

```
mTableID VARCHAR2 (256);
```

```
curr_year varchar2 (4);
```

```
curr_month varchar2 (2);
```

```
BEGIN
```

```
AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE started.');
```

```
PAX_FORMS_UTL.HIDETOOLBAR (ParAXForm);
```

```
ParAXForm('TOOLBAR.KEY_COMMIT').ENABLED:='TRUE';
```

```
SETFORM(paraxform);
```

```
select count(*) into cnt from VX_CL_LOG;
```

```
paraxform('SYSBLOCK.TOPIC').val := '<b>'||PAX_LANG.getmsg('Přehled  
jízď', 'CZ')||'</b> ';
```

```
-- ('||PAX_LANG.getmsg('Počet záznamů', 'CZ')||': '||cnt||');  
AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE ended.');
```

```
select to_char(sysdate, 'YYYY') into curr_year from dual;
```

```
paraxform('CONTROL.DATE_FROM').val := curr_year;
```

```
-- CONTROL_DATE_FROM (paraxform);
```

```
-- CONTROL_DATE_TO (paraxform);
```

```
PAX_FORMS_UTL.setBlockReset(ParAXForm, 'VX_CL_LOG');
```

```
PAX_FORMS_UTL.reinitFormTable(ParAXForm, 'VX_CL_LOG');
```

```
paraxform('CONTROL.CAR').val := null;
```

```
control_car (paraxform);
```

```
END WHEN_NEW_FORMS_INSTANCE;  
-----  
-
```

```
-- TOOLBAR  
-----  
-
```

```
procedure KEY_NEW (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS)AS
```

```
newCompId varchar2(256);
```

```
editFormName varchar2(256) := 'F_E_VX_CL_LOG';
```

```
parReadOnly varchar2(5) := 'false';
```

```
BEGIN
```

```

-- select id into pid from VX_CL_CARS where rowid = prow;
axlogdebug('car: ' || paraxform ('CONTROL.CAR').val);

newCompId:=PAX_FORMS_WIN.CREATEFORMWINDOW(editFormName,PAX_LANG.getmsg('Nový
záznam','CZ'),'true',null, PAX_AXTABLE.GetComponentID(ParAXForm, 'VX_CL_LOG'));
PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_CAR', paraxform
('CONTROL.CAR').val);
PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_ID', null);
PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId, 'PAR_READ_ONLY', parReadOnly);
END KEY_NEW;

procedure KEY_EDIT (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
prow rowid;
BEGIN
select rowid into prow from VX_CL_LOG where
ROWIDTOCHAR(rowid)=paraxform('VX_CL_LOG.AXCURRENTROW').val;
detail(paraxform,prow);
END KEY_EDIT;

procedure KEY_PRINT (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
l_par pax_reports.t_rep_param;
IDCAR number := ParAXForm('CONTROL.CAR').Val;
P_YEAR number := ParAXForm('CONTROL.DATE_FROM').Val;
P_MONTH varchar2 (10) := ParAXForm('CONTROL.DATE_TO').Val;
BEGIN
if p_month = 00 then
PAX_FORMS.dialogOk(ParAXForm, 'Není vybráný konkrétní měsíc.','Chyba');
else
l_par('IDCAR'):= IDCAR;
l_par('P_YEAR'):= P_YEAR;
l_par('P_MONTH'):= P_MONTH;
paraxform('TOOLBAR.KEY_PRINT').val := pax_reports.getreportJ(28, l_par);
end if;
END KEY_PRINT;

procedure KEY_HELP (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
v_help_file varchar2(600) := '/Dokumentace/Administrace systému/Kniha
jízdy.pdf';
file_id number;
BEGIN
select id into file_id from vx_dirdata where pvx_files.get_full_name(id) =
v_help_file;
PAX_FORMS_UTL.openURL(ParAXForm, 'filevx/' || file_id, false);
END KEY_HELP;

procedure KEY_SETTINGS ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS,pBlock in
varchar2) as
begin
PAX_AXTABLE.showColumnEditor(PAX_AXTABLE.GetComponentID(ParAXForm,pBlock));
exception when others then
AXSTACKTRACE;
end;

procedure KEY_XLSX ( parAXForm IN OUT NOCOPY AXFORMS.TYPE_ITEMS, pBlock in
varchar2, parBlobData OUT BLOB, parFileContentType OUT VARCHAR2, parFileName
OUT VARCHAR2 ) as
vQuery VARCHAR2(31024);

```

```

vDatasource varchar2(255) := pBlock;
idc varchar2(32000) := PAX_AXTABLE.GetComponentID(ParAXForm, vDatasource);
visibleColumns VARCHAR2(32767);
begin
  visibleColumns := PAX_AXTABLE.getVisibleColumnsList(idc);
  visibleColumns := replace (visibleColumns, 'AX__ROWID,', '');
  visibleColumns := replace (visibleColumns, 'DETAIL,', '');
  vQuery := 'SELECT ' || visibleColumns || ' FROM ( ' || PAX_PAGE.getServerPar
(idc, 'dataQueryNoFormat') || ')';
  --vQuery := pax_page.getserverpar (PAX_AXTABLE.GetComponentID(ParAXForm,
'QAKCE'), 'dataQueryNoFormat');
  axLogDebug('sqlSource: ' || vQuery);
  DBMS_LOB.CreateTemporary(parBlobData, TRUE);
  axlogdebug('***** GENEXCEL START
*****');
  parblobData := PAX_OFFICE.gen_excel(vQuery, vDatasource);
  axlogdebug('***** GENEXCEL END
*****');
  parFileContentType:='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet';
  parFileName:=vDatasource||'.xlsx';
exception when others then
  AXSTACKTRACE;
end;
-----
-
-- FORM METHODS
-----
-
procedure DETAIL (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS, prow rowid,
pReadOnly in varchar2 default 'false') as
  newCompId varchar2(256);
  lockTable varchar2(256) := 'VX_CL_LOG';
  editFormName varchar2(256) := 'F_E_VX_CL_LOG';
  pId number;
  vChyba varchar2(20000);
  parReadOnly varchar2(5) := pReadOnly;
  pstatus varchar2(256);
begin
  axlogdebug('Procedure DETAIL started. ');
  select id into pid from VX_CL_LOG where rowid = prow;
  select status into pstatus from VX_CL_LOG where rowid = prow;

  if pid is null then
    pax_forms.dialogok(paraxform, PAX_LANG.getmsg('Vyberte
záznam', 'CZ'), PAX_LANG.getmsg('Chyba', 'CZ'), 2);
    return;
  end if;

  if pstatus = 'SM' then
    parReadOnly := 'true';
  else
    parReadOnly := 'false';
  end if;

  if parReadOnly = 'true' then
    pax_forms.dialogok(paraxform, 'Záznam schválen - jen pro
čtení', PAX_LANG.getmsg('Informace', 'CZ'), 4);

```

```

        newCompId:= pax_forms_win.showDetail
(paraxform,prow,lockTable,editFormName,parReadOnly);
    else
        newCompId:= pax_forms_win.showDetail
(paraxform,prow,lockTable,editFormName,parReadOnly);
    end if;

    PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId,'PAR_ID',pid);
    PAX_FORMS_WIN.ADDFORMPARAMETER(newCompId,'PAR_READ_ONLY',parReadOnly);
    PAX_PAGE.setServerPar(newCompId||'win','onCloseRefreshComponent',
PAX_AXTABLE.GetComponentID(ParAXForm,'VX_CL_LOG'));
    axlogdebug('Procedure DETAIL ended.');
```

**end DETAIL;**

```

procedure CONTROL_CAR (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
pid number;
mTableID varchar2 (100);
pname varchar2 (256);
begin
--select name into pname from qax_users;

If instr(PAX_USERS.getAuthorization('T', PAX_PAGE.getCURtransaction(),
null),'D')>0 then
    declare
        cursor c is select ID,
                        long_name
                        from vx_cl_cars
                        order by long_name;
        r c%rowtype;
    begin
        PAX_FORMS_UTL.RemoveItemList(paraxform, 'CONTROL.CAR');
        for r in c loop
            PAX_FORMS_UTL.AddItemList(paraxform, 'CONTROL.CAR', r.ID,
r.long_name, r.ID);
        end loop;
    end ;
end if;

If instr(PAX_USERS.getAuthorization('T', PAX_PAGE.getCURtransaction(),
null),'W')>0 then
    declare
        cursor c is select ID,
                        long_name
                        from vx_cl_cars
                        where ID_USER = PAX_USERS.GetLoggedID
                        order by long_name;
        r c%rowtype;
    begin
        PAX_FORMS_UTL.RemoveItemList(paraxform, 'CONTROL.CAR');
        for r in c loop
            PAX_FORMS_UTL.AddItemList(paraxform, 'CONTROL.CAR', r.ID,
r.long_name, r.ID);
        end loop;
    end ;
end if;

-- if instr(PAX_USERS.getAuthorization('T', PAX_PAGE.getCURtransaction(),
null),'R')>0 then

```



```

load_date (paraxform);

end CONTROL_CAR;

procedure CONTROL_DATE_FROM (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
begin
  load_date (paraxform);

end CONTROL_DATE_FROM;

procedure CONTROL_DATE_TO (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
begin
  load_date (paraxform);

end CONTROL_DATE_TO;

procedure load_date (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
mTableID VARCHAR2 (256);
sum_dist number;
qw varchar(2000);
qw2 varchar (2000);
pcar varchar (256);
p_year varchar (4);
p_month varchar (2);
curr_year varchar (4);
begin
select to_char(sysdate, 'YYYY') into curr_year from dual;

p_year:= paraxform('CONTROL.DATE_FROM').val;
p_month:= paraxform('CONTROL.DATE_TO').val;
pcar := ParAXForm('CONTROL.CAR').Val;

if p_year is not null and p_month = 00 then
  qw := 'to_char(LOG_DATE, 'YYYY') = '|| p_year ||';
  elsif p_year is null then
  qw := 'to_char(LOG_DATE, 'YYYY') <= '|| curr_year ||';
  else
  qw := 'to_char(LOG_DATE, 'YYYY') = '|| p_year ||' AND
to_char(LOG_DATE, 'MM') = '|| p_month ||';
end if;

if ParAXForm('CONTROL.CAR').Val is not null then
  qw2 := 'id_car = ' || pcar ;
  else
  qw2 := 'id_car = 0';
end if;

axlog('to_char(LOG_DATE, 'YYYY') = '|| p_year ||' AND to_char(LOG_DATE, 'MM')
= '|| p_month ||' and id_car = '|| ParAXForm('CONTROL.CAR').Val);
axlog (qw || ' and id_car = '|| ParAXForm('CONTROL.CAR').Val);

--FILTR TABULKY
mTableID := PAX_AXTABLE.GetComponentID(ParAXForm, 'VX_CL_LOG');
PAX_AXTABLE.setWhereClause(mTableID, qw || ' and ' || qw2 );

----SUMA VZDALENOSTI

```

```

-- select SUM(DISTANCE) into sum_dist from vx_cl_log where (log_date BETWEEN
paraxform('CONTROL.DATE_FROM').val AND paraxform('CONTROL.DATE_TO').val) and
ID_CAR = paraxform('CONTROL.CAR').val;
-- paraxform('CONTROL.SUM').val := sum_dist;
end load_date;

```

```

procedure SETFORM ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS ) as
  mTableID varchar2(256);
begin
  --veci co maji byt pristupne vzdy dostanou forceenable true
  --veci co maji byt pristupne pouze pokud ma uzivatel pristup na zapis
  dostanou enabled true
  --veci co nemaji byt pristupne dostanou enabled false
  --VNITRNI TOOLBAR
  paraxform('SYSBLOCK.KEY_NEW').enabled:='true';
  paraxform('SYSBLOCK.KEY_EDIT').enabled:='true';
  paraxform('SYSBLOCK.KEY_EDIT').forceenable:='true';
  paraxform('SYSBLOCK.KEY_DELETE').enabled:='false';
  paraxform('SYSBLOCK.KEY_COMMIT').enabled:='false';
  paraxform('SYSBLOCK.KEY_HELP').enabled:='true';
  paraxform('SYSBLOCK.KEY_HELP').forceenable:='true';
  paraxform('SYSBLOCK.KEY_HISTORY').enabled:='true';
  paraxform('SYSBLOCK.KEY_HISTORY').forceenable:='true';
  paraxform('SYSBLOCK.KEY_PRINT').enabled:='true';
  paraxform('SYSBLOCK.KEY_SETTINGS').enabled:='true';
  paraxform('SYSBLOCK.KEY_SETTINGS').forceenable:='true';
  paraxform('SYSBLOCK.KEY_XLSX').enabled:='true';
  paraxform('SYSBLOCK.KEY_XLSX').forceenable:='true';
end;

```

```

END "F_P_VX_CL_LOG";

```

```

create or replace PACKAGE BODY "F_E_VX_CL_LOG" AS

```

```

PRAGMA SERIALLY_REUSABLE;

```

```

-----
-
-- WHEN NEW FORMS INSTANCE
-----
-

```

```

procedure WHEN_NEW_FORMS_INSTANCE (ParAXForm in out nocopy
PAX_FORMS.TYPE_ITEMS) AS
mTableID varchar2 (150);
BEGIN
  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE started. ');
  PAX_FORMS_UTL.HIDETOOLBAR (ParAXForm);
  if GETPAR(paraxform,'par_read_only') = 'false' then
    ParAXForm('TOOLBAR.KEY_COMMIT').ENABLED:='TRUE';
  end if;

  paraxform('VX_CL_LOG.ID').val := GETPAR(paraxform,'par_id');
  paraxform('VX_CL_LOG.ID_CAR').val := GETPAR(paraxform,'par_car');

  PAX_FORMS.AXREFRESH(paraxform);
  PAX_FORMS_UTL.setBlockReset(ParAXForm, 'VX_CL_LOG');
  PAX_FORMS_UTL.reinitFormTable(ParAXForm, 'VX_CL_LOG');

```

```

PAX_FORMS_UTL.setBlockReset(ParAXForm, 'VX_CL_CARS');
PAX_FORMS_UTL.reinitFormTable(ParAXForm, 'VX_CL_CARS');

PAX_FORMS_UTL.setBlockReset(ParAXForm, 'VX_CL_FUEL_LOG');
PAX_FORMS_UTL.reinitFormTable(ParAXForm, 'VX_CL_FUEL_LOG');

LOG_USE(ParAXForm);
AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE ended.');
```

```
END WHEN_NEW_FORMS_INSTANCE;
```

```

-----
-
-- TOOLBAR
-----
-

```

```

procedure KEY_COMMIT (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
BEGIN
```

```

    if paraxform('VX_CL_LOG.ID').val is null then
        paraxform('VX_CL_LOG.ID').val := pax_forms.axinsert (paraxform,
'VX_CL_LOG.ID');
        commit;
```

```

pax_page.setserverpar(PAX_FORMS_UTL.GETCOMPONENTID(paraxform), 'par_id', paraxform('VX_CL_LOG.ID').val);
```

```

    else
        pax_forms.axupdate(paraxform);
        commit;
    end if;
```

```

    if paraxform('VX_CL_LOG.LOG_DATE').val is null then
        PAX_FORMS.dialogOk(ParAXForm, 'Není zadané datum.', 'Chyba');
    END IF;
```

```

    PAX_FORMS.SETINACTIVE(paraxform);
END KEY_COMMIT;
```

```

procedure KEY_ROLLBACK (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
BEGIN
```

```

    rollback;
    NULL;
END KEY_ROLLBACK;
```

```

procedure KEY_REFRESH (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
BEGIN
```

```

    NULL;
    rollback;
END KEY_REFRESH;
```

```

-----
-
-- FORM METHODS
-----
-

```

```

procedure BUTT_NEW (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
NewRowID rowid;
id_car VARCHAR2(3);
pid VARCHAR2(3);
BEGIN
id_car := paraxform('VX_CL_LOG.ID_CAR').val;
pid := paraxform('VX_CL_LOG.ID').val;
INSERT
  INTO VX_CL_FUEL_LOG
  (
    ID,
    ID_CAR,
    ID_LOG,
    AMOUT,
    PRICE,
    MILEAGE,
    LOG_NOTE
  )
VALUES
  (
    null,
    id_car,
    pid,
    null,
    null,
    null,
    null
  );

commit;
END BUTT_NEW;

procedure BUTT_DEL (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS, prow rowid)
as
begin
  delete from VX_CL_FUEL_LOG where rowid=prow;
end;

procedure LOG_USE (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS
begin
  declare
    cursor c is select  short_name,
                       long_name
                   from    vx_codebook
                   where   codebook_type = 'CL_LOG_USE'
                   order by long_name;
    r c%rowtype;
  begin
    PAX_FORMS_UTL.RemoveItemList(paraxform, 'VX_CL_LOG.LOG_USE');
    for r in c loop
      PAX_FORMS_UTL.AddItemList(paraxform, 'VX_CL_LOG.LOG_USE',
r.short_name, r.long_name, r.short_name);
    end loop;
  end;
END LOG_USE;

procedure BUTT_DOPCTI (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) AS

```

```

begin
  PAX_FORMS.DialogPrompt(ParAXForm, 'Celkový počet km -', 40, 'dopocti',
'Dopocti stav km');
end BUTT_DOPOCTI;

END "F_E_VX_CL_LOG";

```

```

create or replace PACKAGE BODY "F_P_VX_CL_OVERVIEW" AS

```

```

PRAGMA SERIALLY_REUSABLE;
procedure load_date (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS);

```

```

-----
-
-- WHEN NEW FORMS INSTANCE
-----
-

```

```

procedure WHEN_NEW_FORMS_INSTANCE (ParAXForm in out nocopy
PAX_FORMS.TYPE_ITEMS) AS

```

```

  cnt number;

```

```

  curr_year varchar2 (4);

```

```

BEGIN

```

```

  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE started. ');
  PAX_FORMS_UTL.HIDETOOLBAR (ParAXForm);
  select count(*) into cnt from VX_CL_LOG;
  paraxform('SYSBLOCK.TOPIC').val := '<b>' || PAX_LANG.getmsg('Přehled
jízd', 'CZ') || '</b> ';

```

```

  select to_char(sysdate, 'YYYY') into curr_year from dual;
  paraxform('CONTROL.DATE_FROM').val := curr_year;
  AXLOGDEBUG('WHEN_NEW_FORMS_INSTANCE ended. ');
  paraxform('CONTROL.CAR').forceenable:='true';
  paraxform('CONTROL.DATE_FROM').forceenable:='true';
  paraxform('CONTROL.DATE_TO').forceenable:='true';
  paraxform('SYSBLOCK.KEY_XLSX').forceenable:='true';

```

```

  PAX_FORMS_UTL.setBlockReset(ParAXForm, 'VX_CL_LOG_VIEW');
  PAX_FORMS_UTL.reinitFormTable(ParAXForm, 'VX_CL_LOG_VIEW');
  control_car (paraxform);

```

```

END WHEN_NEW_FORMS_INSTANCE;

```

```

-----
-
-- TOOLBAR
-----
-

```

```

procedure KEY_XLSX ( parAXForm IN OUT NOCOPY AXFORMS.TYPE_ITEMS, pBlock in
varchar2) as

```

```

begin

```

```

  zslot_utils.key_xlsx2 (paraxform, pblock);

```

```

end;

```

```

procedure KEY_SETTINGS ( ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS,pBlock in
varchar2) as

```

```

begin

```

```

PAX_AXTABLE.showColumnEditor(PAX_AXTABLE.GetComponentID(ParAXForm,pBlock));
end;

```

```

-----
-
-- FORM METHODS
-----
-

```

```

procedure CONTROL_CAR (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
begin

```

```

    declare
        cursor c is select ID,
                           long_name
        from vx_cl_cars
        order by long_name;
    r c%rowtype;
    begin
--      PAX_FORMS_UTL.RemoveItemList(paraxform, 'CONTROL.CAR');
        for r in c loop
            PAX_FORMS_UTL.AddItemList(paraxform, 'CONTROL.CAR', r.ID,
r.long_name, r.ID);
        end loop;
    end ;
load_date (paraxform);
end CONTROL_CAR;

```

```

procedure CONTROL_DATE_FROM (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
begin
    load_date (paraxform);
end CONTROL_DATE_FROM;

```

```

procedure CONTROL_DATE_TO (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
begin
    load_date (paraxform);
end CONTROL_DATE_TO;

```

```

procedure load_date (ParAXForm in out nocopy PAX_FORMS.TYPE_ITEMS) as
mTableID VARCHAR2 (256);

```

```

qw varchar(2000);
qw2 varchar (2000);
pcar varchar (256);
p_year varchar (4);
p_month varchar (2);
curr_year varchar (4);
filtr varchar (200);
begin
select to_char(sysdate, 'YYYY') into curr_year from dual;
p_year:= paraxform('CONTROL.DATE_FROM').val;
p_month:= paraxform('CONTROL.DATE_TO').val;
pcar := ParAXForm('CONTROL.CAR').Val;

```

```

if p_year is not null and p_month = 00 then
    qw := 'to_char(LOG_DATE, 'YYYY') = '|| p_year ||';
    elsif p_year is null then
        qw := 'to_char(LOG_DATE, 'YYYY') <= '|| curr_year ||';
    else

```

```

        qw := 'to_char(LOG_DATE, 'YYYY') = '|| p_year ||' AND
to_char(LOG_DATE, 'MM') = '|| p_month ||';
end if;

if ParAXForm('CONTROL.CAR').Val is null or ParAXForm('CONTROL.CAR').Val = 00
then
    filtr := qw;
else
    qw2 := 'id_car = ' || pcar ;
    filtr := qw || ' and ' || qw2;
end if;

axlog('to_char(LOG_DATE, 'YYYY') = '|| p_year ||' AND to_char(LOG_DATE, 'MM')
= '|| p_month ||' and id_car = '|| pcar);
axlog (qw || ' and id_car = ' || ParAXForm('CONTROL.CAR').Val);
axlog ( filtr);
--FILTR TABULKY
mTableID := PAX_AXTABLE.GetComponentID(ParAXForm, 'VX_CL_LOG_VIEW');
PAX_AXTABLE.setWhereClause(mTableID, filtr);
end load_date;

END "F_P_VX_CL_OVERVIEW";

```

## (C)XML kód reportu

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio version 6.5.1.final using JasperReports
Library version 6.5.1 -->
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="kniha"
pageWidth="595" pageHeight="842" columnWidth="555" leftMargin="20"
rightMargin="20" topMargin="20" bottomMargin="20" uuid="f0008c19-9b51-4355-
9d88-6bdd9c487fdf">
    <property name="com.jaspersoft.studio.data.sql.tables" value=""/>
    <property name="com.jaspersoft.studio.data.sql.SQLQueryDesigner.sash.w1"
value="185"/>
    <property name="com.jaspersoft.studio.data.sql.SQLQueryDesigner.sash.w2"
value="809"/>
    <property name="com.jaspersoft.studio.data.defaultdataadapter"
value="IPZ"/>
    <style name="Table_TH" mode="Opaque" backgroundColor="#FFFFFF">
        <box>
            <pen lineWidth="0.5" lineColor="#000000"/>
            <topPen lineWidth="0.5" lineColor="#000000"/>
            <leftPen lineWidth="0.5" lineColor="#000000"/>
            <bottomPen lineWidth="0.5" lineColor="#000000"/>
            <rightPen lineWidth="0.5" lineColor="#000000"/>
        </box>
    </style>
    <style name="Table_CH" mode="Opaque" backgroundColor="#DEDEDE">
        <box>
            <pen lineWidth="0.5" lineColor="#000000"/>
            <topPen lineWidth="0.5" lineColor="#000000"/>
            <leftPen lineWidth="0.5" lineColor="#000000"/>

```

```

        <bottomPen lineWidth="0.5" lineColor="#000000"/>
        <rightPen lineWidth="0.5" lineColor="#000000"/>
    </box>
</style>
<style name="Table_TD" mode="Opaque" backgroundColor="#FFFFFF">
    <box>
        <pen lineWidth="0.5" lineColor="#000000"/>
        <topPen lineWidth="0.5" lineColor="#000000"/>
        <leftPen lineWidth="0.5" lineColor="#000000"/>
        <bottomPen lineWidth="0.5" lineColor="#000000"/>
        <rightPen lineWidth="0.5" lineColor="#000000"/>
    </box>
</style>
<subDataset name="table_detail2" uuid="ae0bdf1d-9898-4964-8583-41d1cac883ba">
    <property name="com.jaspersoft.studio.data.sql.tables" value=""/>
    <property name="com.jaspersoft.studio.data.defaultdataadapter"
value="IPZ"/>
    <property
name="com.jaspersoft.studio.data.sql.SQLQueryDesigner.sash.w1" value="353"/>
    <property
name="com.jaspersoft.studio.data.sql.SQLQueryDesigner.sash.w2" value="641"/>
    <parameter name="TD_P_ID_CAR" class="java.lang.String"/>
    <parameter name="TD_P_YEAR" class="java.lang.String"/>
    <parameter name="TD_P_MONTH" class="java.lang.String"/>
    <parameter name="TD_SUM_TOTAL_KM_BEGIN"
class="java.math.BigDecimal"/>
    <queryString>
        <![CDATA[SELECT
            ID,
            ID_CAR,
            decode (b_p, 'P', ' ', to_char(LOG_DATE, 'DD. MM. YYYY')) as FORM_DATE,
            LOG_NOTE,
            START_TIME,
            END_TIME,
            LOG_USE,
            decode(LOG_USE, 'PRAC', 'Pracovní', 'JINY', 'Jiný', 'OBCHOD', 'Obchodní',
'SERVIS', 'Servis vozu') AS LOG_USE_TXT,
            DISTANCE,
            decode (b_p, 'P', 0, DISTANCE) as B_DISTANCE,
            decode (b_p, 'P', DISTANCE, 0) as P_DISTANCE,
            B_P
FROM vx_cL_Log
where id_car = ${TD_P_ID_CAR} AND to_char(LOG_DATE, 'YYYY') = ${TD_P_YEAR}
AND to_char (LOG_DATE, 'MM') = ${TD_P_MONTH}
order by b_p asc, log_date, start_time asc]]>
    </queryString>
    <field name="ID" class="java.math.BigDecimal"/>
    <field name="ID_CAR" class="java.math.BigDecimal"/>
    <field name="FORM_DATE" class="java.lang.String">
        <fieldDescription><![CDATA[]]></fieldDescription>
    </field>
    <field name="START_TIME" class="java.lang.String"/>
    <field name="END_TIME" class="java.lang.String"/>
    <field name="LOG_USE" class="java.lang.String"/>
    <field name="DISTANCE" class="java.math.BigDecimal"/>
    <field name="B_P" class="java.lang.String"/>
    <field name="B_DISTANCE" class="java.math.BigDecimal"/>

```



```

        <field name="P_DISTANCE" class="java.math.BigDecimal"/>
        <field name="LOG_NOTE" class="java.lang.String"/>
        <field name="LOG_USE_TXT" class="java.lang.String"/>
        <variable name="sum_distance" class="java.math.BigDecimal"
calculation="Sum">

        <variableExpression><![CDATA[ $\$F\{DISTANCE\}$ ]]></variableExpression>
        </variable>
        <variable name="sum_distance_b" class="java.math.BigDecimal"
calculation="Sum">

        <variableExpression><![CDATA[ $\$F\{B\_DISTANCE\}$ ]]></variableExpression>
        </variable>
        <variable name="sum_distance_p" class="java.math.BigDecimal"
calculation="Sum">

        <variableExpression><![CDATA[ $\$F\{P\_DISTANCE\}$ ]]></variableExpression>
        </variable>
        <variable name="end_sum_km" class="java.lang.Integer">
            <variableExpression><![CDATA[new
Integer( $\$P\{TD\_SUM\_TOTAL\_KM\_BEGIN\}$ .intValue()+ $\$V\{sum\_distance\}$ .intValue())]]></v
ariableExpression>
        </variable>
    </subDataset>
    <parameter name="IDCAR" class="java.lang.String"/>
    <parameter name="P_YEAR" class="java.lang.String"/>
    <parameter name="P_MONTH" class="java.lang.String"/>
    <queryString>
        <![CDATA[select
id,
id_user,
long_name,
reg_plate,
nvl((SELECT SUM(DISTANCE) FROM VX_CL_LOG WHERE ID_CAR =  $\$P\{IDCAR\}$  AND
to_char(log_date, 'YYYY') <  $\$P\{P\_YEAR\}$  ),0) AS TOP_KM_BEGIN_YEAR,
nvl((SELECT SUM(DISTANCE) FROM VX_CL_LOG WHERE ID_CAR =  $\$P\{IDCAR\}$  AND
to_char(log_date, 'YYYY') <=  $\$P\{P\_YEAR\}$  AND to_char (LOG_DATE, 'MM') <
 $\$P\{P\_MONTH\}$ ),0) AS SUM_TOTAL_KM_BEGIN,
(select personal_number from vx_contacts where id_users = id_user) as
person_id,
(select first_name||' '||last_name from ax_users where id = id_user) as
person_name
from
vx_cl_cars
where
id= $\$P\{IDCAR\}$ ]]>
    </queryString>
    <field name="ID" class="java.lang.String"/>
    <field name="ID_USER" class="java.lang.String"/>
    <field name="LONG_NAME" class="java.lang.String"/>
    <field name="REG_PLATE" class="java.lang.String"/>
    <field name="TOP_KM_BEGIN_YEAR" class="java.math.BigDecimal"/>
    <field name="SUM_TOTAL_KM_BEGIN" class="java.math.BigDecimal"/>
    <field name="PERSON_ID" class="java.lang.String"/>
    <field name="PERSON_NAME" class="java.lang.String"/>
    <background>
        <band splitType="Stretch"/>
    </background>

```

```

<title>
  <band height="80" splitType="Stretch">
    <frame>
      <reportElement x="0" y="0" width="555" height="80"
uuid="1dec5b65-8b52-45bb-a8b6-aa29d0e48938">
        <property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
        </reportElement>
        <box topPadding="0" leftPadding="0" rightPadding="0">
          <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
          <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
          <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
          <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <staticText>
          <reportElement x="147" y="11" width="260"
height="20" uuid="9c687690-6043-49c7-b317-075f251c608f"/>
          <textElement textAlignment="Center">
            <font size="14" isBold="true"
pdfEncoding="Cp1250"/>
          </textElement>
          <text><![CDATA[Kniha jízd firemního vozidla
]]></text>
        </staticText>
        <staticText>
          <reportElement x="206" y="63" width="201"
height="13" uuid="66145740-cc7b-42b6-87b4-37954bd376c9"/>
          <textElement textAlignment="Center">
            <font pdfEncoding="Cp1250"/>
          </textElement>
          <text><![CDATA[Garant formuláře: Petra
Andrejsková]]></text>
        </staticText>
        <staticText>
          <reportElement x="9" y="63" width="161"
height="13" uuid="f9b27916-ec9d-4188-a556-8dc34448abb7"/>
          <textElement>
            <font pdfEncoding="Cp1250"/>
          </textElement>
          <text><![CDATA[Vazba na dokument:
S0103]]></text>
        </staticText>
        <staticText>
          <reportElement x="447" y="64" width="105"
height="13" uuid="77e974d0-d1c5-4507-a5a8-c21b5388f68a"/>
          <textElement>
            <font pdfEncoding="Cp1250"/>
          </textElement>
          <text><![CDATA[Označení: F0103 02]]></text>
        </staticText>
      </frame>
    </band>
  </title>
</pageHeader>

```

```

        <band height="7" splitType="Stretch"/>
</pageHeader>
<columnHeader>
    <band height="6" splitType="Stretch"/>
</columnHeader>
<detail>
    <band height="82" splitType="Stretch">
        <staticText>
            <reportElement mode="Opaque" x="321" y="0" width="39"
height="15" bgcolor="#DEDEDE" uuid="849c4b79-e36b-46b3-8710-3b3c08e8bd8f">
                <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
            </reportElement>
            <box leftPadding="0">
                <pen lineWidth="1.0"/>
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            </box>
            <textElement textAlignment="Center"
verticalAlignment="Middle">
                <font pdfEncoding="Cp1250"/>
                <paragraph leftIndent="2"/>
            </textElement>
            <text><![CDATA[SPZ:]]></text>
        </staticText>
        <staticText>
            <reportElement mode="Opaque" x="321" y="18" width="90"
height="15" bgcolor="#DEDEDE" uuid="559fcf19-2f3f-4417-bc2d-03894c2c6fae">
                <property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
            </reportElement>
            <box leftPadding="0">
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            </box>
            <textElement textAlignment="Center"
verticalAlignment="Middle">
                <font pdfEncoding="Cp1250"/>
                <paragraph leftIndent="2"/>
            </textElement>
            <text><![CDATA[Typ vozidla:]]></text>
        </staticText>
        <staticText>
            <reportElement mode="Opaque" x="0" y="61" width="100"
height="15" bgcolor="#DEDEDE" uuid="4398cb40-ff6a-42f3-a96c-fa18948d1b41">

```

```

        <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
        </reportElement>
        <box leftPadding="0">
            <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <textElement textAlignment="Center"
verticalAlignment="Middle">
            <font pdfEncoding="Cp1250"/>
            <paragraph leftIndent="2"/>
            </textElement>
            <text><![CDATA[Jméno řidiče:]]></text>
        </staticText>
        <textField>
            <reportElement mode="Opaque" x="100" y="61"
width="179" height="15" backcolor="#FCF586" uuid="765ab51f-c072-4786-bd36-
e8f9642d2c8e"/>
            <box leftPadding="0">
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            </box>
            <textElement textAlignment="Center"
verticalAlignment="Middle">
                <font isBold="true" pdfEncoding="Cp1250"/>
                <paragraph leftIndent="2"/>
            </textElement>
            <textFieldExpression><![CDATA[ $\$F\{PERSON\_NAME\}$ ]]></textFieldExpression>
        </textField>
        <textField>
            <reportElement x="411" y="18" width="143" height="15"
uuid="652f64e6-dcde-4791-a959-e049087a5d66"/>
            <box leftPadding="0">
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            </box>
            <textElement textAlignment="Center"
verticalAlignment="Middle">
                <font isBold="true" pdfEncoding="Cp1250"/>

```

```

        <paragraph leftIndent="2"/>
    </textElement>

    <textFieldExpression><![CDATA[{$F{LONG_NAME}}]></textFieldExpression>
    </textField>
    <textField>
        <reportElement mode="Opaque" x="360" y="0" width="100"
height="15" bgcolor="#FCF586" uuid="aea3b8f5-3deb-4f5c-a685-3f273f73fc1b"/>
        <box leftPadding="0">
            <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <textElement textAlignment="Center"
verticalAlignment="Middle">
            <font isBold="true" pdfEncoding="Cp1250"/>
            <paragraph leftIndent="2"/>
        </textElement>

    <textFieldExpression><![CDATA[{$F{REG_PLATE}}]></textFieldExpression>
    </textField>
    <staticText>
        <reportElement mode="Opaque" x="321" y="42" width="39"
height="15" bgcolor="#DEDEDE" uuid="b867383d-1c03-47f6-9ff7-1e2a94814c92">
        <property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
        </reportElement>
        <box leftPadding="0">
            <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <textElement textAlignment="Center"
verticalAlignment="Middle">
            <font pdfEncoding="Cp1250"/>
            <paragraph leftIndent="2"/>
        </textElement>
        <text><![CDATA[Rok:]]></text>
    </staticText>
    <staticText>
        <reportElement mode="Opaque" x="411" y="42" width="33"
height="15" bgcolor="#DEDEDE" uuid="05d05596-a39c-4e13-b35f-2bb82b7f0bc3"/>
        <box leftPadding="0">
            <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>

```

```

                                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                </box>
                                <textElement textAlignment="Center"
verticalAlignment="Middle">
                                    <font pdfEncoding="Cp1250"/>
                                    <paragraph leftIndent="2"/>
                                </textElement>
                                <text><![CDATA[Měsíc:]]></text>
                                </staticText>
                                <staticText>
                                    <reportElement mode="Opaque" x="321" y="60" width="90"
height="15" bgcolor="#DEDEDE" uuid="0cb7e70a-72a2-408d-9f15-502b0dd9ad5c"/>
                                    <box leftPadding="0">
                                        <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        </box>
                                    <textElement textAlignment="Center"
verticalAlignment="Middle">
                                        <font pdfEncoding="Cp1250"/>
                                        <paragraph leftIndent="2"/>
                                    </textElement>
                                    <text><![CDATA[Osobní číslo řidiče:]]></text>
                                </staticText>
                                <textField>
                                    <reportElement x="360" y="42" width="51" height="15"
uuid="ec9fa8bf-7187-443c-81d7-7fb53dacc5b5">
                                        <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
                                    </reportElement>
                                    <box>
                                        <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                                        </box>
                                    <textElement textAlignment="Center"
verticalAlignment="Middle">
                                        <font isBold="true" pdfEncoding="Cp1250"/>
                                    </textElement>
                                </textField>
                                <textFieldExpression><![CDATA[ $P_{YEAR}$ ]]></textFieldExpression>
                                </textField>
                                </textField>

```

```

        <reportElement mode="Opaque" x="444" y="42"
width="109" height="15" bgcolor="#FCF586" uuid="7b35d8f3-62b8-4066-9e5f-
efb8519fe44b">
            <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
            <property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
        </reportElement>
        <box>
            <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <textElement textAlignment="Center"
verticalAlignment="Middle">
            <font isBold="true" pdfEncoding="Cp1250"/>
            </textElement>
            <textFieldExpression><![CDATA[ $\$P\{P\_MONTH\}.equals("01")$ 
? "Leden" :
     $\$P\{P\_MONTH\}.equals("02")$  ? "únor" :
     $\$P\{P\_MONTH\}.equals("03")$  ? "březen" :
     $\$P\{P\_MONTH\}.equals("04")$  ? "duben" :
     $\$P\{P\_MONTH\}.equals("05")$  ? "květen" :
     $\$P\{P\_MONTH\}.equals("06")$  ? "červen" :
     $\$P\{P\_MONTH\}.equals("07")$  ? "červenec" :
     $\$P\{P\_MONTH\}.equals("08")$  ? "srpen" :
     $\$P\{P\_MONTH\}.equals("09")$  ? "září" :
     $\$P\{P\_MONTH\}.equals("10")$  ? "říjen" :
     $\$P\{P\_MONTH\}.equals("11")$  ? "listopad" :
    "prosinec"]]></textFieldExpression>
        </textField>
        <textField>
            <reportElement x="411" y="60" width="100" height="15"
uuid="4abc20c2-7682-44f4-acef-63effeefb2ca">
                <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
            </reportElement>
            <box>
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
            </box>
            <textElement textAlignment="Center"
verticalAlignment="Middle">
                <font pdfEncoding="Cp1250"/>
                </textElement>
                <textFieldExpression><![CDATA[ $\$F\{PERSON\_ID\}$ ]]></textFieldExpression>

```

```

        </textField>
    </band>
    <band height="458">
        <componentElement>
            <reportElement x="0" y="16" width="555" height="442"
                uuid="474e7f6f-1c8c-4537-bd20-a3d3cf4afb36">
                <property name="com.jaspersoft.studio.layout"
                    value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
                <property
                    name="com.jaspersoft.studio.table.style.table_header" value="Table_TH"/>
                <property
                    name="com.jaspersoft.studio.table.style.column_header" value="Table_CH"/>
                <property
                    name="com.jaspersoft.studio.table.style.detail" value="Table_TD"/>
            </reportElement>
            <jr:table
                xmlns:jr="http://jasperreports.sourceforge.net/jasperreports/components"
                xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports/components
                    http://jasperreports.sourceforge.net/xsd/components.xsd">
                <datasetRun subDataset="table_detail2"
                    uuid="75289a92-5c1a-40ff-a8f7-ad1f4f8d4fb8">
                    <datasetParameter name="TD_P_ID_CAR">
                        <datasetParameterExpression><![CDATA[{$F{ID}}]></datasetParameterExpression>
                    </datasetParameter>
                    <datasetParameter name="TD_P_YEAR">
                        <datasetParameterExpression><![CDATA[{$P{P_YEAR}}]></datasetParameterExpression>
                    </datasetParameter>
                    <datasetParameter name="TD_P_MONTH">
                        <datasetParameterExpression><![CDATA[{$P{P_MONTH}}]></datasetParameterExpression>
                    </datasetParameter>
                    <datasetParameter name="TD_SUM_TOTAL_KM_BEGIN">
                        <datasetParameterExpression><![CDATA[{$F{SUM_TOTAL_KM_BEGIN}}]></datasetParameterExpression>
                    </datasetParameter>
                    <connectionExpression><![CDATA[{$P{REPORT_CONNECTION}}]></connectionExpression>
                </datasetRun>
                <jr:column width="70" uuid="bf771a66-18f7-4e7b-af7f-f98ff0826f0b">
                    <property
                        name="com.jaspersoft.studio.components.table.model.column.name"
                        value="Column1"/>
                    <property
                        name="com.jaspersoft.studio.unit.height" value="pixel"/>
                    <jr:tableHeader style="Table_TH"
                        height="0" rowSpan="1">
                        <property
                            name="com.jaspersoft.studio.layout"
                            value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>

```



```

                                <property
name="com.jaspersoft.studio.unit.height" value="px"/>
                                </jr:tableHeader>
                                <jr:columnHeader style="Table_CH"
height="20" rowspan="1">
                                <staticText>
                                <reportElement x="0" y="0"
width="70" height="20" uuid="0894744c-3672-4b21-9225-41a41f574074"/>
                                <textElement
textAlignment="Center" verticalAlignment="Middle">
                                <font
pdfEncoding="Cp1250"/>
                                </textElement>
                                <text><![CDATA[Datum]]></text>
                                </staticText>
                                </jr:columnHeader>
                                <jr:columnFooter style="Table_CH"
height="20" rowspan="1"/>
                                <jr:detailCell style="Table_TD"
height="30">
                                <textField isBlankWhenNull="true">
                                <reportElement x="0" y="0"
width="70" height="30" uuid="cd8f48e7-f6f0-4e6a-9eaf-f7b222dea83f"/>
                                <textElement
textAlignment="Center" verticalAlignment="Middle">
                                <font
pdfEncoding="Cp1250"/>
                                <paragraph
leftIndent="2"/>
                                </textElement>
                                <textFieldExpression><![CDATA[$F{FORM_DATE}]]></textFieldExpression>
                                </textField>
                                </jr:detailCell>
                                </jr:column>
                                <jr:column width="210" uuid="0aa42014-76dc-
456f-b55b-866abf21ad2c">
                                <property
name="com.jaspersoft.studio.components.table.model.column.name"
value="Column2"/>
                                <jr:tableHeader style="Table_TH"
height="0" rowspan="1">
                                <property
name="com.jaspersoft.studio.layout"
value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
                                <property
name="com.jaspersoft.studio.unit.height" value="px"/>
                                </jr:tableHeader>
                                <jr:columnHeader style="Table_CH"
height="20" rowspan="1">
                                <staticText>
                                <reportElement x="0" y="0"
width="210" height="20" uuid="b5f6222d-433e-4dee-86e5-2e70d49b7e5e">
                                <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
                                </reportElement>

```

```

                                <textElement
textAlignment="Center" verticalAlignment="Middle">
                                <font
pdfEncoding="Cp1250"/>
                                </textElement>
                                <text><![CDATA[Místo počátku
- cíl - místo ukončení cesty]]></text>
                                </staticText>
                                </jr:columnHeader>
                                <jr:columnFooter style="Table_CH"
height="20" rowSpan="1">
                                <staticText>
                                <reportElement x="0" y="0"
width="210" height="20" uuid="63b68c88-45eb-48c4-9f63-24afbd90b25c"/>
                                <textElement
verticalAlignment="Middle">
                                <font isBold="true"
pdfEncoding="Cp1250"/>
                                <paragraph
leftIndent="5"/>
                                </textElement>
                                <text><![CDATA[Celkem]]></text>
                                </staticText>
                                </jr:columnFooter>
                                <jr:detailCell style="Table_TD"
height="30">
                                <textField isBlankWhenNull="true">
                                <reportElement x="0" y="0"
width="210" height="30" uuid="5a6eab70-86fa-4f15-86c2-44e610fc5f7f"/>
                                <textElement
textAlignment="Center" verticalAlignment="Middle"/>
                                <textFieldExpression><![CDATA[{$F{LOG_NOTE}}]></textFieldExpression>
                                </textField>
                                </jr:detailCell>
                                </jr:column>
                                <jr:columnGroup width="80" uuid="bcc8a585-744e-
42d5-bb35-060350c57957">
                                <property
name="com.jaspersoft.studio.components.table.model.column.name" value="Columns
[2]"/>
                                <jr:tableFooter style="Table_TH"
height="20" rowSpan="1">
                                <box>
                                <pen lineWidth="2.0"/>
                                <topPen lineWidth="2.0"
                                <leftPen lineWidth="2.0"
                                <bottomPen lineWidth="2.0"
                                <rightPen lineWidth="2.0"
                                </box>
                                <textField>

```

```

<reportElement mode="Opaque"
x="0" y="0" width="80" height="20" bgcolor="#BAC3FF" uuid="2895b1f1-1c4c-
45f6-b45e-11c16fc4223a"/>
<textElement
textAlignment="Center" verticalAlignment="Middle"/>
<textFieldExpression><![CDATA[ $\$P\{TD\_SUM\_TOTAL\_KM\_BEGIN\}$ ]]></textFieldExpr
ession>
</textField>
</jr:tableFooter>
<jr:column width="40" uuid="95600ec4-
5857-46f7-bdb2-174464f41036">
<property
name="com.jaspersoft.studio.components.table.model.column.name"
value="Column3"/>
<jr:tableHeader style="Table_TH"
height="0" rowspan="1">
<property
name="com.jaspersoft.studio.layout"
value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
<property
name="com.jaspersoft.studio.unit.height" value="px"/>
</jr:tableHeader>
<jr:tableFooter style="Table_TH"
height="0" rowspan="1"/>
<jr:columnHeader style="Table_CH"
height="20" rowspan="1">
<staticText>
<reportElement x="0"
y="0" width="40" height="20" uuid="35d06031-d8cd-4252-be9d-3c64ab63aa4b"/>
<textElement
textAlignment="Center" verticalAlignment="Middle">
<font
pdfEncoding="Cp1250"/>
</textElement>
<text><![CDATA[Od]]></text>
</staticText>
</jr:columnHeader>
<jr:columnFooter style="Table_CH"
height="20" rowspan="1"/>
<jr:detailCell style="Table_TD"
height="30">
<textField
isBlankWhenNull="true">
<reportElement x="0"
y="0" width="40" height="30" uuid="0d5f9fce-790f-4155-a981-5c76892dd2a7"/>
<textElement
textAlignment="Center" verticalAlignment="Middle">
<font
pdfEncoding="Cp1250"/>
<paragraph
leftIndent="2"/>
</textElement>
<textFieldExpression><![CDATA[ $\$F\{START\_TIME\}$ ]]></textFieldExpression>
</textField>
</jr:detailCell>

```

```

        </jr:column>
        <jr:column width="40" uuid="9d0a5a52-
314f-463b-8a20-f7dce9cb0d37">
            <property
name="com.jaspersoft.studio.components.table.model.column.name"
value="Column4"/>
            <jr:tableHeader style="Table_TH"
height="0" rowSpan="1">
                <property
name="com.jaspersoft.studio.layout"
value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
                <property
name="com.jaspersoft.studio.unit.height" value="px"/>
            </jr:tableHeader>
            <jr:tableFooter style="Table_TH"
height="0" rowSpan="1"/>
            <jr:columnHeader style="Table_CH"
height="20" rowSpan="1">
                <staticText>
                    <reportElement x="0"
y="0" width="40" height="20" uuid="d183ab3a-a347-4944-be55-eac961361e12"/>
                    <textElement
textAlignment="Center" verticalAlignment="Middle">
                        <font
pdfEncoding="Cp1250"/>
                    </textElement>
                </staticText>
            </jr:columnHeader>
            <jr:columnFooter style="Table_CH"
height="20" rowSpan="1"/>
            <jr:detailCell style="Table_TD"
height="30">
                <textField
isBlankWhenNull="true">
                    <reportElement x="0"
y="0" width="40" height="30" uuid="38e4ba35-d6e6-4200-892c-957c0191f119"/>
                    <textElement
textAlignment="Center" verticalAlignment="Middle">
                        <font
pdfEncoding="Cp1250"/>
                    <paragraph
leftIndent="2"/>
                </textElement>
            </textField>
            <jr:detailCell>
                <textFieldExpression><![CDATA[ $\$F\{END\_TIME\}$ ]]></textFieldExpression>
            </textField>
            </jr:detailCell>
        </jr:column>
    </jr:columnGroup>
    <jr:columnGroup width="106" uuid="f2d47bf7-
2b9a-4693-8d8e-26791ad08996">
        <property
name="com.jaspersoft.studio.components.table.model.column.name" value="Columns
[2]"/>
        <jr:tableFooter style="Table_TH"
height="20" rowSpan="1">

```

```

                <textField>
                    <reportElement mode="Opaque"
x="0" y="0" width="106" height="20" backcolor="#FCF586" uuid="f2446c9f-ab4d-
4216-9686-f6de5c29b609"/>
                <box>
                    <pen
lineWidth="2.0"/>
                    <topPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    <leftPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    <bottomPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    <rightPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                </box>
                <textElement
textAlignment="Center" verticalAlignment="Middle">
                    <font isBold="true"
pdfEncoding="Cp1250"/>
                    <paragraph
leftIndent="2"/>
                </textElement>
                <textFieldExpression><![CDATA[ $\sum_{distance}$ ]]></textFieldExpression>
                </textField>
                </jr:tableFooter>
                <jr:column width="56" uuid="a91bca6c-
da1d-4bea-84b8-f642ac14b657">
                    <property
name="com.jaspersoft.studio.components.table.model.column.name"
value="Column5"/>
                    <jr:tableHeader style="Table_TH"
height="0" rowSpan="1">
                        <property
name="com.jaspersoft.studio.layout"
value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
                        <property
name="com.jaspersoft.studio.unit.height" value="px"/>
                    </jr:tableHeader>
                    <jr:tableFooter style="Table_TH"
height="0" rowSpan="1"/>
                    <jr:columnHeader style="Table_CH"
height="20" rowSpan="1">
                        <staticText>
                            <reportElement x="0"
y="0" width="56" height="20" uuid="53e84add-05b0-41cb-ac98-ffb7ff1d1e5f"/>
                            <textElement
textAlignment="Center" verticalAlignment="Middle">
                                <font
pdfEncoding="Cp1250"/>
                            </textElement>
                        </staticText>
                    </jr:columnHeader>
                    <jr:columnFooter style="Table_CH"
height="20" rowSpan="1">
                <text><![CDATA[Služebně]]></text>
            </jr:columnFooter>

```

```

isBlankWhenNull="true">
mode="Opaque" x="0" y="0" width="56" height="20" bgcolor="#FCF586"
uuid="b9c38547-e14a-49a6-a3bb-655371e9768b"/>
lineWidth="2.0"/>
textAlignment="Center" verticalAlignment="Middle">
isBold="true" pdfEncoding="Cp1250"/>
leftIndent="2"/>
<textFieldExpression><![CDATA[ $\sum_{b}$ ]]></textFieldExpression>
</jr:columnFooter>
<jr:detailCell style="Table_TD"
height="30">
isBlankWhenNull="true">
y="0" width="56" height="30" uuid="ffe04a89-af1a-4986-83be-929843cabb9a"/>
textAlignment="Center" verticalAlignment="Middle">
pdfEncoding="Cp1250"/>
leftIndent="2"/>
<textFieldExpression><![CDATA[ $B_{DISTANCE}$ ]]></textFieldExpression>
</jr:detailCell>
</jr:column>
<jr:column width="50" uuid="3bd73a52-
b7c9-4981-ae83-acf6d961a35b">
name="com.jaspersoft.studio.components.table.model.column.name"
value="Column6"/>
height="0" rowspan="1">
name="com.jaspersoft.studio.layout"
value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
name="com.jaspersoft.studio.unit.height" value="px"/>
height="0" rowspan="1"/>
height="20" rowspan="1">
<staticText>
y="0" width="50" height="20" uuid="7b625255-8dfc-4617-bec0-9632a1e79006"/>

```

```

textAlignment="Center" verticalAlignment="Middle">
pdfEncoding="Cp1250"/>
</textElement>
</font>
</textElement>
<text><![CDATA[Soukromě]]></text>
</staticText>
</jr:columnHeader>
<jr:columnFooter style="Table_CH"
height="20" rowSpan="1">
isBlankWhenNull="false">
mode="Opaque" x="0" y="0" width="50" height="20" backgroundColor="#FCF586"
uuid="235bf1b5-eb6a-4710-8a5a-f27997b309d3"/>
<reportElement
<box>
<pen
lineWidth="2.0"/>
</box>
<textField
textAlignment="Center" verticalAlignment="Middle">
isBold="true" pdfEncoding="Cp1250"/>
leftIndent="2"/>
<paragraph
</textElement>
<textFieldExpression><![CDATA[ $\sum_{p}$ ]]></textFieldExpression>
</textField>
</jr:columnFooter>
<jr:detailCell style="Table_TD"
height="30">
<textField
isBlankWhenNull="true">
y="0" width="50" height="30" uuid="eb69c13a-3fc6-4fc0-aecb-3b118a811295"/>
<reportElement x="0"
<textField
textAlignment="Center" verticalAlignment="Middle">
pdfEncoding="Cp1250"/>
leftIndent="2"/>
<paragraph
</textElement>
<textFieldExpression><![CDATA[ $P_{DISTANCE}$ ]]></textFieldExpression>
</textField>
</jr:detailCell>
</jr:column>
</jr:columnGroup>
<jr:column width="89" uuid="a9b6cd6f-e735-499b-
8017-612b705935d3">
<property
name="com.jaspersoft.studio.components.table.model.column.name"
value="Column7"/>
<jr:tableHeader style="Table_TH"
height="0" rowSpan="1">

```

```

                    <property
name="com.jaspersoft.studio.layout"
value="com.jaspersoft.studio.editor.layout.VerticalRowLayout"/>
                    <property
name="com.jaspersoft.studio.unit.height" value="px"/>
                    </jr:tableHeader>
                    <jr:tableFooter style="Table_TH"
height="20" rowSpan="2">
                    <textField isBlankWhenNull="true">
                    <reportElement mode="Opaque"
x="0" y="0" width="89" height="20" backcolor="#FCF586" uuid="1527af14-9f8f-
4798-af47-2e67f4cf08b7"/>
                    <box>
                    <pen
lineWidth="2.0"/>
                    <topPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    <leftPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    <bottomPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    <rightPen
lineWidth="2.0" lineStyle="Solid" lineColor="#000000"/>
                    </box>
                    <textElement
textAlignment="Center" verticalAlignment="Middle">
                    <font isBold="true"
pdfEncoding="Cp1250"/>
                    <paragraph
leftIndent="2"/>
                    </textElement>
                    <textFieldExpression><![CDATA[ $\{end\_sum\_km\}$ ]]></textFieldExpression>
                    </textField>
                    </jr:tableFooter>
                    <jr:columnHeader style="Table_CH"
height="20" rowSpan="1">
                    <staticText>
                    <reportElement x="0" y="0"
width="89" height="20" uuid="c6b14b87-6e14-4cc4-bebc-5b80435d9c06">
                    <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
                    </reportElement>
                    <textElement
textAlignment="Center" verticalAlignment="Middle">
                    <font
pdfEncoding="Cp1250"/>
                    </textElement>
                    <text><![CDATA[Účel
jízdy]]></text>
                    </staticText>
                    </jr:columnHeader>
                    <jr:columnFooter style="Table_CH"
height="20" rowSpan="1"/>
                    <jr:detailCell style="Table_TD"
height="30">
                    <textField isBlankWhenNull="true">

```



```

width="89" height="30" uuid="70a77c9f-467f-43a1-b7ee-4c26ce688aa4"/>
textAlignment="Center" verticalAlignment="Middle">
pdfEncoding="Cp1250"/>
leftIndent="2"/>
</textElement>
</textElement>
</textField>
</jr:detailCell>
</jr:column>
</jr:table>
</componentElement>
<staticText>
<reportElement mode="Opaque" x="360" y="1" width="106"
height="15" bgcolor="#DEDEDE" uuid="f3343076-38d9-4d52-bbaf-df4d5d6592ad">
<property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
<property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
</reportElement>
<box>
<topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
<leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
<bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
<rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
</box>
<textElement textAlignment="Center"/>
<text><![CDATA[Ujeto km]]></text>
</staticText>
</band>
</detail>
<columnFooter>
<band height="1" splitType="Stretch"/>
</columnFooter>
<pageFooter>
<band height="104" splitType="Stretch">
<staticText>
<reportElement x="0" y="0" width="250" height="74"
uuid="87daaa7b-ef4d-418f-afb0-2eac29ce9bbb"/>
<box>
<topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
<leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
<bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
<rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
</box>
<textElement textAlignment="Left">

```

```

        <font pdfEncoding="Cp1250"/>
        <paragraph lineSpacingSize="1.0"
leftIndent="2"/>
        </textElement>
        <text><![CDATA[Poznámky:]]></text>
</staticText>
<staticText>
        <reportElement x="400" y="0" width="156" height="74"
uuid="c35faa07-4f7b-42af-b282-941942715b1f">
        <property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
        </reportElement>
        <box>
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <textElement textAlignment="Left">
        <font pdfEncoding="Cp1250"/>
        <paragraph leftIndent="2"/>
        </textElement>
        <text><![CDATA[Schválil:

Datum:]]></text>
        </staticText>
<staticText>
        <reportElement x="250" y="0" width="150" height="74"
uuid="3e65db06-a977-4470-9584-1ffe42c1561e">
        <property
name="com.jaspersoft.studio.unit.height" value="pixel"/>
        <property
name="com.jaspersoft.studio.unit.width" value="pixel"/>
        </reportElement>
        <box>
                <topPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <leftPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <bottomPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
                <rightPen lineWidth="1.0" lineStyle="Solid"
lineColor="#000000"/>
        </box>
        <textElement textAlignment="Left">
        <font pdfEncoding="Cp1250"/>
        <paragraph leftIndent="2"/>
        </textElement>
        <text><![CDATA[Podpis řidiče:

```

```

Datum:]]></text>
</staticText>
<staticText>
  <reportElement x="0" y="74" width="250" height="29"
  uuid="08cec521-c411-4716-a51c-9b063490b25c"/>
  <box>
    <topPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
    <leftPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
    <bottomPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
    <rightPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
  </box>
  <textElement textAlignment="Left">
    <font pdfEncoding="Cp1250"/>
    <paragraph leftIndent="2"/>
  </textElement>
  <text><![CDATA[Účel služební jízdy: Pracovní
  Jiný
  Obchodní Servis vozu]]></text>
</staticText>
<textField isBlankWhenNull="true">
  <reportElement x="250" y="74" width="306" height="29"
  uuid="150d178f-4e15-4b10-a9af-30423740d6cb">
    <property
    name="com.jaspersoft.studio.unit.height" value="pixel"/>
    <property
    name="com.jaspersoft.studio.unit.width" value="pixel"/>
  </reportElement>
  <box>
    <topPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
    <leftPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
    <bottomPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
    <rightPen lineWidth="1.0" lineStyle="Solid"
    lineColor="#000000"/>
  </box>
  <textElement>
    <font pdfEncoding="Cp1250"/>
    <paragraph leftIndent="2"/>
  </textElement>
  <textFieldExpression><![CDATA["Stav ujetých km na
  počátku roku (zařazení do provozu): "
  + ${TOP_KM_BEGIN_YEAR} +"\n" +
  "Stav ujetých km na konci roku:"]]></textFieldExpression>
</textField>
</band>
</pageFooter>
<summary>
  <band height="23" splitType="Stretch"/>
</summary>
</jasperReport>

```

## (D) Kód databázových tabulek a pohledů

-----  
-- VX\_CL\_LOG  
-----

```
CREATE TABLE vx_cl_log
( ID NUMBER NOT NULL ENABLE,
  ID_car NUMBER,
  log_date DATE,
  start_time DATE,
  end_time DATE,
  log_note VARCHAR2(70 BYTE),
  distance NUMBER,
  b_p CHAR (2 BYTE),
  log_use VARCHAR2(50 BYTE),
  ID_projects NUMBER,
  status CHAR (2 BYTE) DEFAULT 'RO',
  ID_documents NUMBER,
  CONSTRAINT vx_cl_log_PK PRIMARY KEY (ID)
CONSTRAINT VX_CL_CARS_FK FOREIGN KEY (ID_car)
REFERENCES car (ID)
CONSTRAINT projects_FK FOREIGN KEY (ID_projects)
REFERENCES projects (ID)
CONSTRAINT documents_FK FOREIGN KEY (ID_documents)
REFERENCES documents (ID)
);

COMMENT ON COLUMN vx_cl_log.ID IS ' ';
COMMENT ON COLUMN vx_cl_log.ID_car IS ' ';
COMMENT ON COLUMN vx_cl_log.log_date IS 'Datum';
COMMENT ON COLUMN vx_cl_log.start_time IS 'Čas startu';
COMMENT ON COLUMN vx_cl_log.end_time IS 'Čas cíle';
COMMENT ON COLUMN vx_cl_log.log_note IS 'Popis';
COMMENT ON COLUMN vx_cl_log.distance IS 'Vzdálenost';
COMMENT ON COLUMN vx_cl_log.b_p IS 'B - služební, P - soukromé';
COMMENT ON COLUMN vx_cl_log.log_use IS 'Účel jízdy ';
COMMENT ON COLUMN vx_cl_log.ID_projects IS 'Projekt';
COMMENT ON COLUMN vx_cl_log.status IS 'RO - rozpracování, SM - schváleno';
COMMENT ON COLUMN vx_cl_log.ID_documents IS 'Úkol';
/
CREATE SEQUENCE vx_cl_log_ID MINVALUE 1 MAXVALUE 9999999999999999999999999999
INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE;
/
CREATE OR REPLACE TRIGGER vx_cl_log_trg BEFORE INSERT ON VX_CL_LOG
FOR EACH ROW
  WHEN (NEW.ID IS NULL) BEGIN
SELECT vx_cl_log_ID.NEXTVAL INTO :NEW.ID FROM DUAL;
EXCEPTION WHEN others THEN
ROLLBACK
END;
/
ALTER TRIGGER vx_cl_log_TRG ENABLE;
```

-----  
-- VX\_CL\_CARS  
-----





```

CREATE OR REPLACE FORCE VIEW qvx_cl_log_overview AS
SELECT
  a.rowid AS ax____rowid,
  a.id,
  a.id_car,
  a.distance,
  a.log_note,
  a.log_date,
  a.start_time || ' - ' || a.end_time AS a_time,
  DECODE(B_P, 'B', 'Služební', 'P', 'Pracovní') AS b_p_txt,
  DECODE(LOG_USE, 'PRAC', 'Pracovní', 'JINÝ', 'Jiný', 'OBCHOD', 'Obchodní',
'SERVIS', 'Servis vozu') AS log_use_txt,
  pax_users.getinfo (c.id_user) AS user_name,
  c.reg_plate,
  c.long_name,
  SUM(b.price) AS S_price,
  SUM(b.amout) AS s_amout
FROM vx_cl_log a
LEFT OUTER JOIN vx_cl_fuel_log b ON a.id = b.id_log
INNER JOIN vx_cl_cars c ON a.id_car=c.id
GROUP BY a.rowid, a.id, a.id_car, a.distance, a.log_note, c.id_user,
c.reg_plate, c.long_name, a.start_time, a.end_time, a.log_date, a.b_p,
a.log_use
ORDER BY a.log_date DESC;

```

## (E) Uživatelská příručka k formuláři registr vozidel

### Obecné ovládaní

#### Panel nástrojů

Každý formulář obsahuje panel nástrojů





#### Význam tlačítek

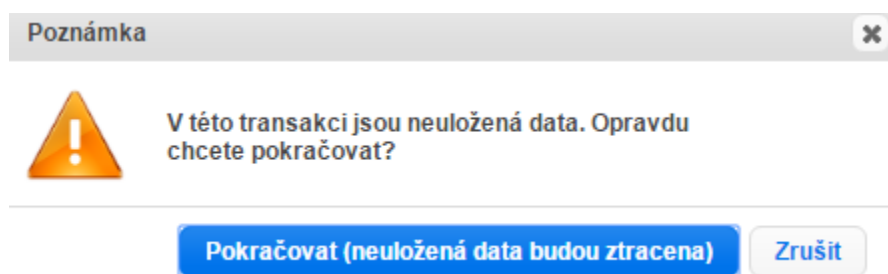
- Nový záznam (jen pro přehledy)
- Editace (zobrazení detailu)
- Odstranění záznamu
- Uložení (jen pro editaci)
- Nápověda
- Zobrazení historie změn
- Tisk
- Export do Excelu (jen pro přehledy)
- Výběr zobrazení sloupců (jen pro přehledy)

### Formuláře

Formuláře jsou buď přehledové, nebo editační. Přehledový formulář obsahuje panel nástrojů, tabulku s přehledem údajů a případně další akční tlačítka.

Editační se většinou zobrazí jako okno s formulářem s údaji k vyplnění/zobrazení. Okno lze zavřít  nebo klávesou Esc. Formulář obsahuje i záložky, kliknutím na záložku se zobrazí údaje z dané záložky.

Data se ukládají stiskem ikony  pro uložení v panelu nástrojů. Při uzavření okna zkontroluje systém, zda jsou nějaké neuložené změny. Pokud ano, zeptá se.



Pokud zvolíte Pokračovat, systém data neuloží a formulář uzavře. Pokud zvolíte zrušit, vrátíte se do formuláře a můžete data uložit.





## Přehled formuláře

Administrace AX > Kniha jízd > Evidence vozů (2017) v. 1.00



Přehled vozidel (Počet záznamů: 2)

	SPZ	Název	Rok výroby	Průměrná spotřeba	Velikost nádrže	Zodpovědný pracovník
1	test2	testtest				Bc. Petra Andrejsková
2	TEST	TEST2	2000			Bc. Petra Andrejsková

Přehled vozidel se standardně ovládá pomocí panelu nástrojů (viz kapitola Panel nástrojů)

Tlačítkem  zobrazíte detail vozidla, který můžete editovat. To samé uděláte i tlačítkem  v panelu nástrojů.

### Detail vozidla

Kliknutím na tlačítko  v panelu nástrojů či na tlačítko  v řádku formuláře u daného vozidla, zobrazíte tuto tabulka:

Detail vozidla v. 1.00


ID

SPZ  Rok výroby


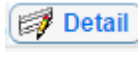
Název

Velikost nádrže  Průměrná spotřeba

Zodpovědný pracovník  🔍

V té můžete změnit nebo zadat SPZ, rok výroby, název, velikost nádrže a průměrnou spotřebu. Vyplnění SPZ je povinné. Kliknutím na tlačítko  u pole „zodpovědný pracovník“ vyberete ze zobrazeného seznamu pracovníka, který bude přiřazen k vozidlu.

## Založení nového záznamu

Pro založení nového záznamu použijte tlačítko  v přehledovém formuláři. Otevře se stejný formulář jako při kliknutí na tlačítko , ale nebude obsahovat žádná data. Povinné pole, které **musí** být vyplněné je SPZ. Formulář se ovládá stejně jako editace formuláře vozidla (viz Detail vozidla).