

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Oracle Application Express**  
Bakalářská práce

Autor: Petr Medek  
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Barbora Tesařová, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou/diplomovou práci zpracoval/zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 20.4.2018

*vlastnoruční podpis*

Petr Medek

Poděkování:

Děkuji vedoucí bakalářské práce Ing. Barboře Tesařové, Ph.D. za metodické vedení práce a cenné připomínky.

## **Anotace**

Tato bakalářská práce se zabývá popisem nástroje pro vývoj webových aplikací s názvem Oracle Application Express. V první, teoretické části je objasněn pojem APEX, včetně jeho technologií, nad kterými pracuje a poté je rozebrán postup zprovoznění (s instalací, nebo bez ní). Následuje objasnění důležitých pojmů, které se v tomto vývojovém prostředí vyskytují a popis modulů tvořící jeho funkcionalitu. Druhá část práce obsahuje tvorbu výukové aplikace, která je opět rozdělena na moduly, kde každý z nich představuje téma, které bývá pro studenty problematické.

## **Annotation**

### **Title: Oracle Application Express**

This bachelor thesis deals with a description of Oracle Application Express application development tool. The first, theoretical part explains the concept of APEX, including its technologies, beyond which it works and then disassembles the process of commissioning (with or without the installation). The following is an explanation of important concepts that occur in this development environment and a description of the modules that make up its functionality. The second part of the thesis includes the creation of a learning application, which is divided into modules again. Each of them presents a topic that is problematic for students.

## Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	Teoretická část .....	3
3.1	Co je APEX .....	3
3.2	Popis požadavků na systém (Instalace a konfigurace) .....	5
3.2.1	Instalace Oracle Database .....	5
3.2.2	Instalace Oracle Application Express.....	6
3.2.3	Zprovoznění vývojového prostředí bez instalace .....	9
3.3	Popis pojmů vývojového prostředí .....	9
3.3.1	Instance administration, workspaces a schémata .....	10
3.4	Moduly APEX.....	13
3.4.1	Application builder .....	14
3.4.2	SQL Workshop .....	16
3.4.3	Team Development.....	17
3.4.4	Packaged Apps.....	18
4	Tvorba výukové aplikace .....	19
4.1	Rozbor aplikačních modulů.....	19
4.1.1	Modul 1. – Založení nové aplikace .....	20
4.1.2	Modul 2. – Práce s obrázky v tabulkách.....	26
4.1.3	Modul 3. – Stylizace šablony .....	27
4.1.4	Modul 4. – Vkládání dat do více tabulek z jednoho formuláře .....	30
4.1.5	Modul 5. – Security (bezpečnost).....	33
4.1.6	Modul 6. – Sessions .....	38
4.1.7	Modul 7. – Vložení tlačítka do řádku tabulky .....	39
4.1.8	Závěrem o výukové aplikaci .....	41

5	Shrnutí výsledků .....	42
6	Závěry a doporučení.....	43
7	Seznam použité literatury.....	44
8	Seznam obrázků.....	48

# 1 Úvod

V dnešní moderní době, kde vládnu především mobilní zařízení, pokročila i tvorba webových stránek. Obyčejný uživatel si to neuvědomuje, ale v mnoha případech již nejde o jednoduché prezentační weby, které pouze zobrazují statická data, ale interaktivní aplikace, u kterých je nutno čerpat data z databází (jednoduchým příkladem může být ověřování uživatelů pro přihlašování, nebo načítání komentářů pro články). A právě tento typ aplikací umožňuje APEX<sup>1</sup> vytvářet. Pro nekomerční účely je licence na vytvořený software zadarmo, a pokud se jedná o jednodušší program, tak na něj vývojář ani nepotřebuje mít znalosti programovacích jazyků. Je to z toho důvodu, že Oracle připravil pro tuto situaci příjemného průvodce, kde je možno řadu věcí navolit, což přináší i další výhodu, týkající se rychlého vývoje webového softwaru. Všechny tyto plusy, které APEX nabízí, vedly ke vzniku této práce, která se týká jeho objasnění, zprovoznění a popisu, plus rozbor problematických částí, během tvorby výukové aplikace.

---

<sup>1</sup> Zkratka pro Oracle Application Express, což je webově zaměřené vývojové prostředí, které běží na Oracle database.

## 2 Cíl práce

Smyslem této bakalářské práce je vytvoření výukových materiálů zájemcům o Oracle APEX, především pro studenty předmětu Databázové systémy.

První, teoretická část řeší několik otázek – nejprve je nutno čtenáře seznámit s pojmem APEX. Další problematickou částí je postup instalace a zprovoznění, včetně požadavků na systém, na kterém toto vývojové prostředí bude pracovat. Po uvedení APEXu do chodu by se v něm měl vývojář umět zorientovat, z čehož vyplývá nutnost porozumět klíčovým pojmům a rozebrat jednotlivé moduly, ze kterých je složen.

Pro druhou část práce jsou stanoveny okruhy, které byly pro studenty problematické. Z každého z nich se vytvoří modul, jejichž množina bude tvořit výukovou aplikaci. Avšak i její webové stránky budou vyvíjeny v samotném Oracle APEX.



## 3 Teoretická část

V této části bude čtenář seznámen s pojmem APEX, včetně jeho technologií a jazyky, se kterými pracuje. Poté se popíší možnosti jeho zprovoznění, spolu s podrobným postupem, včetně nutných požadavků kladených na systém. Po uvedení tohoto vývojového prostředí do chodu budou objasněny důležité pojmy a následně rozebrány jeho jednotlivé moduly.

### 3.1 Co je APEX

Application Express je webově orientované vývojové prostředí (tedy k vývoji stačí pouze okno uživatelského prohlížeče) běžící nad systémem řízení báze dat<sup>2</sup> (dále jen SŘBD) společnosti Oracle a samo o sobě je i webovou aplikací<sup>3</sup>. Jeho první vydání pochází z roku 2004, kterým byla verze 1.5. Avšak tato aplikace měla ještě jiný název a to Oracle HTML DB. Z hlediska jejího účelu je určena především pro vývoj webových aplikací, se zaměřením na jednoduchost a rychlost tvorby. S pomocí APEX lze i lehce, pomocí šablonového systému tvořit pěkný, responzivní design. V případě menších aplikací nejsou ani potřeba znalosti technologií či jazyků, se kterými toto vývojové prostředí pracuje. Těmi jsou: [14] [15]

- **HTML** – ve zkratce HyperText Markup Language, je základní jazyk, kterým jsou tvořeny webové stránky. Text je vkládán mezi značky (proto Markup), kterým se říká tagy. Bez HTML by prohlížeč nevěděl, jak má zobrazit obsah, případně jiné elementy, jako jsou obrázky, videa, atd. [11]
- **CSS** – kaskádové styly (Cascading Style Sheets) jsou určeny k pokročilému formátování internetových stránek. Díky nim lze aplikovat skupinové stylování HTML elementů stejného druhu, čímž si vývojář velmi usnadňuje práci. Potom

---

<sup>2</sup> Systém řízení báze dat (SŘBD) je software, který zajišťuje práci s databází a celkově je spravuje (např. vytváří, upravuje, maže).

<sup>3</sup> Webová aplikace je taková, která se zprostředkovává webovým serverem přes internet (tedy přes uživatelského prohlížeče). Pro využívání takových aplikací není nutno instalací, či aktualizací do klientského počítače.

je i tedy například možno přizpůsobit vzhled stránek různým rozlišením displejů. V dnešní době se už v podstatě nevyskytují webové stránky, které by kromě HTML nepoužívaly i kaskádové styly. [4] [16]

- **Javascript** - programovací jazyk, který je nedílnou součástí všech úspěšných, moderních webových stránek. Lze jím oživit web (např. tvorbou animací), vytvářet aplikace uvnitř stránek (třeba grafické hodiny, hry...), ale používá se i třeba pro validaci formulářů. Pracuje na straně klienta (tedy na zařízení uživatele, který prohlíží dané stránky), tím pádem nedokáže sám získat data ze serveru a zároveň lze i vypnout, což znamená, že se na něj vývojář nemůže spoléhat (hlavně v oblastech zabezpečení). Další jeho nevýhodou je, že nedokáže pracovat se soubory na disku kromě cookies<sup>4</sup>. [5] [1]
- **AJAX (Asynchronous JavaScript and XML)** - způsob programování v Javascriptu, který se vyznačuje tím, že zvládne měnit vzhled obsahu stránky bez nutnosti jejího kompletního překreslení a znovunačtení (tedy obnoví se jen obsah nějakého elementu uvnitř). Je tomu docíleno díky nové schopnosti Javascriptu komunikovat se serverem a asynchronnosti zpracování, tedy není nutno čekat na jeho odpověď, ale pokračuje se v provádění kódu, takže může uživatel dál pokračovat ve své práci na stránce. Pozdější synchronizace je provedena po přijetí odpovědi tzv. callbackem - následnou funkcí. [13]

---

<sup>4</sup> HTTP cookies jsou nízkoobjemová data, která zasílá server do prohlížeče. Ten je následně uloží a může využívat pro potřeby daného webu (např. nese informaci o přihlášení určitého uživatele).

- **PL/SQL** – ve zkratce Procedural Language/Structured Query Language, je nadstavbou jazyka SQL (Structured Query Language) od společnosti Oracle, která pracuje s daty v databázi (získává, ukládá, nebo je obsluhuje). V této podobě už nevykonává jen dotazy, ale je obohacena o programovací funkce (např. možnost psaní procedur - viz bod 4.1.7). Ve vývojovém prostředí, o kterém je tato práce, hraje důležitou roli. Samotné jádro APEXu totiž tvoří set balíčků v Oracle database, s názvem PL/SQL Web Toolkit, který díky svým procedurám generuje HTML stránky. [17] [14] [18]

## **3.2 Popis požadavků na systém (Instalace a konfigurace)**

Než se vývojář pustí do práce s tímto nástrojem, tak by měl vědět, jakým způsobem ho uvést do provozu a jaké požadavky jsou kladeny na systém, kde se bude nacházet. Pro vývoj webových aplikací v APEXu jsou na výběr celkem dvě možnosti – prostředí si sám vývojář nainstaluje do SŘBD Oracle (Oracle Database), nebo s ním bude pracovat přes webový server apex.oracle.com, kde je nutná pouze registrace vývojáře a nastavení pracovního prostředí (tedy bez nutnosti jakékoliv instalace). Níže budou popsány oba způsoby zprovoznění.

### **3.2.1 Instalace Oracle Database**

Již bylo řečeno, že instalace APEXu je nutná v případě, pokud vývojář chce, aby byl tento program spojen s jeho zavedenou Oracle database. V následující situaci bude tedy předpokládáno, že si programátor chce nainstalovat Oracle Application Express, ale dosud u sebe ještě nemá tento SŘBD zavedený. Výhodou instalace na lokální disk je to, že lze aplikace vyvíjet offline, tedy není potřeba mít přístup k internetu a zároveň se tím vyhne časové prodlevě při komunikaci se serverem Oracle.

Ze všeho nejdřív je však nejprve potřeba splnit minimální systémové požadavky a až poté se lze pustit do samotné instalace:

- **Paměť RAM:** 1-2 GB (v závislosti na operačním systému. Windows 7 - 1 GB, Windows 8 – 2 GB)
- **Procesor:** AMD64, nebo Intel Extended memory (EM64T)
- **Volné místo na disku:** 5.39 GB
- **Video adapter:** 256 barev
- **Rozlišení obrazovky:** 1024 x 768

Pokud je vše v pořádku, tak si vývojář stáhne ze stránek Oracle, v sekci **downloads Oracle Database 11g Release 2**. Budou to dva zabalené soubory, které rozbalí někam na disk do stejné složky. Poté přes soubor **setup.exe** postupně v několika krocích projde instalací.

V části s názvem **Installation option** zvolí **create and configure database**, protože jinak by konfiguraci zbytečně vynechal a také zatím nemá žádnou verzi tohoto software, kterou by chtěl aktualizovat (položka **Install database software only**), čímž odpadá i třetí možnost (**Upgrade an existing database**).

Krok, který se nazývá **System class**, dává vývojáři na výběr ze dvou tříd. První z nich (kterou si vybere) je označena jako **Desktop class**. Tuto možnost zvolí, pokud si Oracle database instaluje na svůj laptop, nebo stolní počítač. Druhou třídou je **Server class**, která se týká instalace na servery (využití především v produkčních datových centrech). Je zde obsaženo více konfiguračních možností.

Ve fázi **Typical installation** nechá všechny cesty (základní cesta k SŘBD, další části Oracle a umístění database souboru) tak, jak jsou nastaveny v základu, ale zvolí si zde administrátorské heslo pro privilegovaný databázový účet.

V posledních krocích už jen zkontroluje, zda vše správně zadal a poté začne instalační proces, po kterém by mělo být vše úspěšně splněno. [7]

### 3.2.2 Instalace Oracle Application Express

Po zprovoznění Oracle Database na uživatelském zařízení lze začít s přípravami na instalaci APEXu. Opět se zde budou muset splnit určité instalační požadavky a samotná instalace bude i o něco náročnější:

- **Oracle database:** instalace je možná až od verze 9.2.0.3, nebo vyšší
- **Požadavky prohlížeče:** (viz bod 3.2.1)
- **Volné místo na disku:** 694 MB (plus 34 MB za každý dodatečný jazyk)

Pokud vše vyhovuje, tak na stránkách **oracle.com**, v sekci **downloads** je možno nalézt Oracle Application Express, kde se stáhne balík s názvem **Oracle Application Express 5.0.3 - English language only** (případně novější verzi) a ten je nutno rozbalit někam u sebe na disk, do adresáře s názvem **apex** (pozor! Název musí být shodný, jinak instalace nebude fungovat), nejlépe např. **C:\apex**, aby se to vývojáři dobře hledalo.

Nyní bude potřeba si zapnout příkazovou řádku pomocí příkazu **cmd** a přes něj se dostat do svého adresáře. V případě cesty **C:\apex** se zadá **cd C:\apex**.

Další krok je již zkouškou, zda si programátor nainstaloval Oracle database. Následně se pokračuje v psaní do příkazové řádky. Příkaz **sqlplus** by měl zapnout interaktivní dotazovací nástroj nazývaný **SQL\*Plus** pro komunikaci se SŘBD.

Pokud vše proběhlo v pořádku, tak vývojář vyzván k zadání uživatelského jména. Následně zadá **/ as sysdba**, čímž bude přihlášen jako správce (díky tomuto oprávnění lze vykonávat speciální akce, jako je například zapnutí, či vypnutí SŘBD).

Samotná instalace je zahájena zadáním příkazu **@apexins sysaux sysaux temp /i/**. Tento proces zabere i několik desítek minut.

Opět je potřeba zapnout SQL\*plus, ale tentokrát rovnou s přihlášením. Zadá se tedy **sqlplus / as sysdba**.

Pro správnou funkčnost vývojového prostředí nutno změnit sadu obrázků, které se zobrazují přes webové prostředí APEX. To je provedeno příkazem **@apxldimg C:\** (zde stačí zadat jen disk, na kterém se nachází APEX, protože příkaz obsahuje metodu, díky které nalezne potřebnou složku).

Nyní je nutno napsat **@apxconf**. Tímto skriptem se umožní nakonfigurovat port pro Oracle XML DB HTTP server a navolit si název a heslo pro účet Instance administratora (viz bod 3.3.1) APEXu.

Poslední věc, kterou je ještě potřeba udělat, se týká zprovoznění veřejného a anonymního účtu, k možnému přihlášení prostřednictvím webového prohlížeče.

Toho je docíleno zadáním příkazů v nástroji SQL\*plus **alter user apex\_public\_user account unlock;** a **alter user anonymous account unlock;**

V nadcházejících krocích budou použity pojmy jako workspace, schémata a Instance administration (včetně pojmů, co se k nim vztahují), které jsou vysvětleny v bodě 3.3.1. Zde by měl již vývojář pracovat s instancí samotného vývojového prostředí, přesněji se sekcí s názvem **Oracle Application Express Administration Services** (neboli Instance administration), kde bude moct být založen Workspace (pracovní prostor pro budoucí vývoj obsahující uživatele, aplikace a jejich správ, atd.), který bude pracovat s databázovým schématem, což je logická struktura používané databáze. Vývojář si zapne webový prohlížeč a zadá adresu [http://localhost:8080/apex/apex\\_admin](http://localhost:8080/apex/apex_admin) (pozor, číslo portu je zde uvedeno jako výchozí, tedy doplní se číslo, které bylo nastaveno v odstavci výše). Mělo by se mu objevit přihlašovací okénko, kde vloží své administrátorské údaje a bude přeměrován do administrátorského prostředí.

V této sekci je nutno zvolit **Create Workspace**. V prvním kroku se zobrazí formulář pro zadání unikátního jména pracovního prostředí, dále ID (identifikační číslo - jedinečný, číselný, kladný identifikátor větší než 100000) a nějaký popis tohoto Workspace. Druhý krok je určen schématu. V tuto chvíli vývojář žádné nemá (lze zde vybrat již existující) a bude vytvořeno nové, proto se nechají všechna pole, tak jak jsou přednastavena. Případně je možno v posledním poli, s názvem **Space Quota (MB)**, vyhradit více místa, pokud bude budoucí struktura databáze většího rozsahu. Následně APEX vytvoří nové schéma. Třetí krok se týká založení administrátora (přesněji Workspace administrátora) pro správu nového pracovního prostředí. Po zadání potřebných údajů vývojář v posledním kroku zkontroluje, zda vše sedí a bude mu vytvořen Workspace.

Finální krok obsahuje zadání adresy <http://localhost:8080/apex> do prohlížeče, vyplnění přihlašovacích údajů pro Workspace administrátora a pokud vše proběhlo v pořádku, tak se vývojář ocitne v pracovním prostředí. [8]

### 3.2.3 Zprovoznění vývojového prostředí bez instalace

Vzhledem k tomu, že tato práce je určena především studentům, případně nadšence se zájmem o APEX, tak zde bude zmíněn i postup, kterým lze rychle zpřístupnit vývoj databázových aplikací, ovšem pouze pro testovací účely. Je to z toho důvodu, že se pracuje s nízkým pracovním prostorem (do 25 MB) a pokud není vývojář aktivní, tak mu v pravidelných cyklech hrozí, že bude jeho Workspace smazán (vždy nutno potvrdit aktivitu přes doručený email). [20] [21]

#### Požadavky prohlížeče

Vzhledem k tomu, že prostředí běží ve webovém prohlížeči, je nutno splňovat určitá kritéria k vyvíjení aplikací. Prohlížeč musí podporovat JavaScript, HTML 4.0 a CSS 1.0 standardy. Dále musí následující prohlížeče splňovat tyto požadavky:

- Microsoft Internet Explorer 6.0, nebo novější
- Firefox 1.0, nebo novější

#### Postup zprovoznění

Po splnění předchozích požadavků vývojář otevře webovou stránku apex.oracle.com. Zde si zažádá o vytvoření nového Workspace, s vyhrazeným prostorem na Oracle cloudu, založí uživatelský účet a po obdržení emailu o úspěšném přidělení může začít pracovat.

### 3.3 Popis pojmů vývojového prostředí

Během práce s vývojovým prostředím, by měl vývojář rozumět několika základním pojmům, které se v APEXu vyskytují. Jedná se o Instance administration (správu instance), workspaces a také schémata. Důležité jsou ovšem i role uživatelů, co vystupují napříč aplikací. Mezi hlavní patří tyto 4 skupiny:

- **Instance administrator (správce instance)** – je takzvaný superuživatel, který přiděluje a spravuje jmenné prostory a celou hostovanou instanci APEX. Je to

jediná z rolí, která nejde při vytváření účtu v pracovním prostoru uživateli přidělit.

- **Workspace administrator (správce p. prostoru)** – provádí úkony, které se týkají údržby daného workspace, tedy správa uživatelských účtů, monitorování jejich aktivit, případně prohlížení logů<sup>5</sup>
- **Developer (vývojář)** – uživatel co vyvíjí aplikace. Je potřeba u něj nastavit, zda bude mít přístup k modulu Application builder a SQL Workshop. Může být vybráno obojí, nebo aspoň jeden z nich.
- **End user (konečný uživatel)** – může pouze pracovat s databází, nebo aplikacemi, ale nemá přístup do samotného workspace a tedy ani k vývojářským nástrojům [8]

### 3.3.1 Instance administration, workspaces a schémata

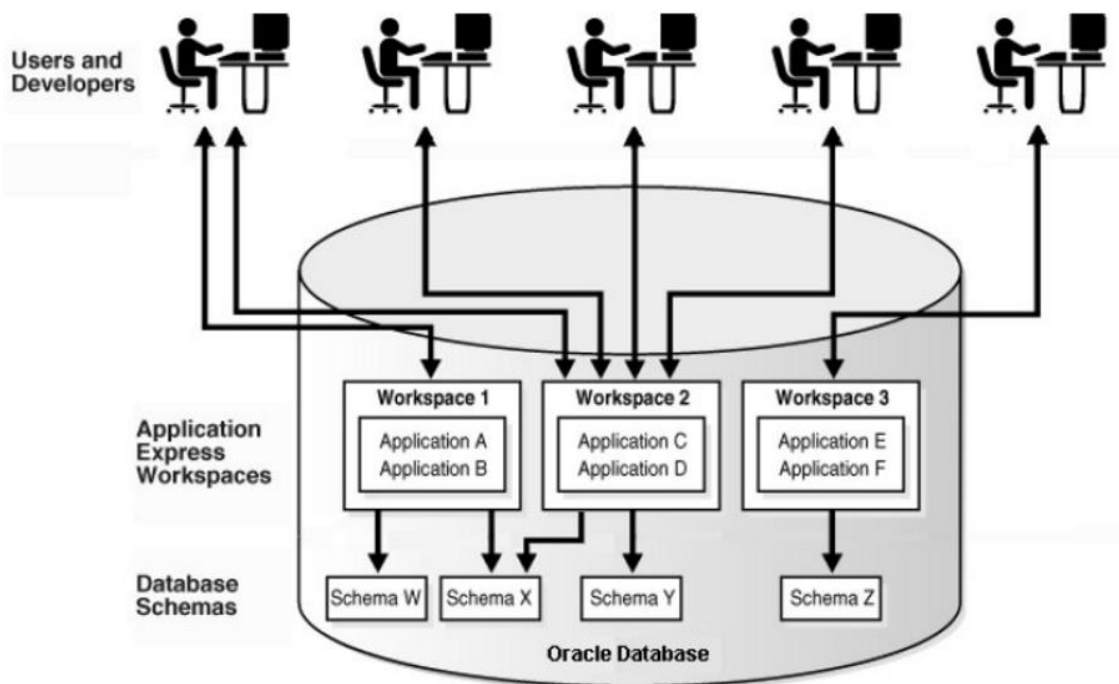
To, co by vývojář nazval hlavní částí Oracle APEX, se nazývá workspace. Tím se myslí libovolná skupinu projektů na jednom místě, případně ještě uložené nějaké nastavení. S těmito prostory pracují jednotliví **uživatelé, či vývojáři**. Ti se mohou dle libosti přihlásit k jakémukoliv z nich (pokud na nich tedy mají založený uživatelský účet). Zavedením workspace má tu výhodu, že jsou všechny rozpracované projekty pohromadě (tedy lze mezi nimi pohodlně přepínat).

Případně také mohou tyto jednotlivé pracovní prostory společně sdílet databázové objekty (např. tabulky, klíče, sekvence,... atd.) díky tzv. **schématům**. [2]

---

<sup>5</sup> Soubor, který programy vytvářejí pro záznam informací o svých činnostech a běhu. Slouží k rozpoznání chyb, které případně nastaly.

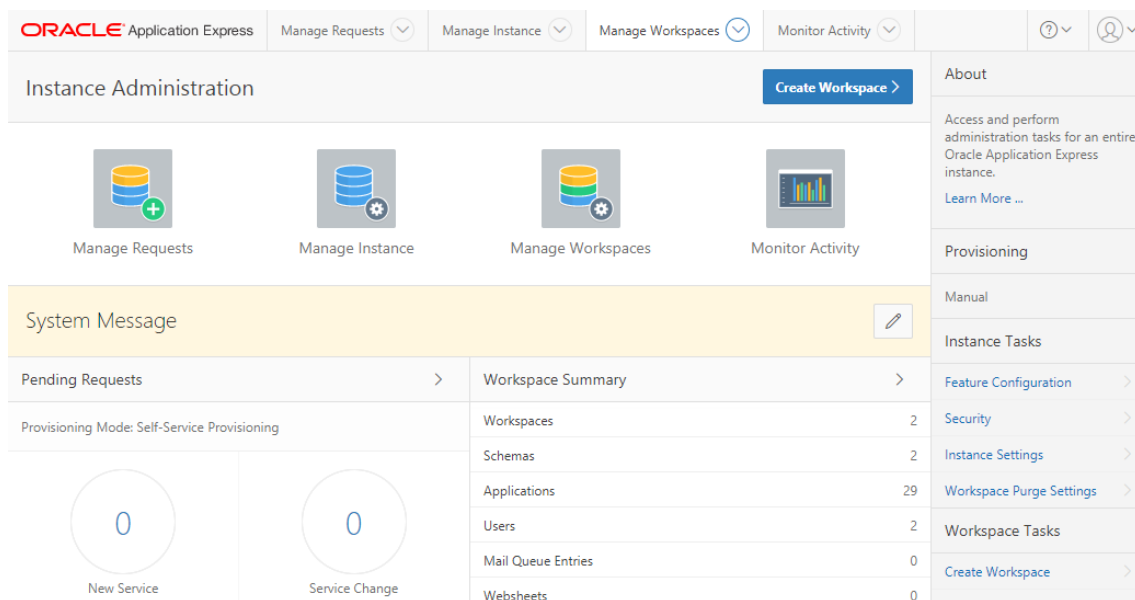




Obr. 1 vztah mezi uživateli, pracovními prostory a databázovými schématy.

Zdroj: [3]

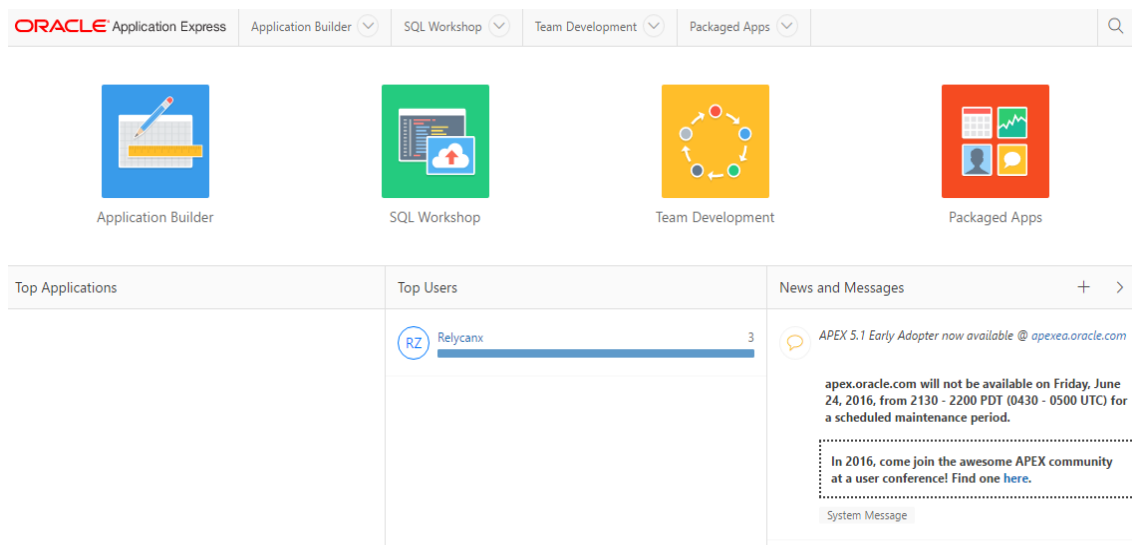
Nakonec je tu otázka, jak vlastně vznikají jednotlivá pracovní prostředí. Vzhledem k předchozímu tématu týkající se instalace, Oracle Application Express obsahuje



**Obr. 2 ukázka prostředí Instance Administration**

prostředí s názvem Instance administration (tedy Správa instance), kde se jednotlivá workspaces spravují a je i možnost monitorovat jejich činnost. Do této části má ovšem přístup jen uživatel s rolí Instance administrator.

## 3.4 Moduly APEX



**Obr. 3 hlavní stránka s moduly**

Vývojové prostředí APEX obsahuje celkem 4 moduly, se kterými se bude v průběhu tvorby pracovat. K vývoji je nezbytný první, s názvem Application builder. Dále se tu nachází takový, který usnadňuje práci – SQL<sup>6</sup> Workshop a další dva (Team development a Packaged Apps) jsou naopak využity jen někdy, nebo případně vůbec. Postupně bude popsán každý z nich. [10]

---

<sup>6</sup> dotazovací jazyk, pomocí kterého se provádí operace nad relační databází

### 3.4.1 Application builder

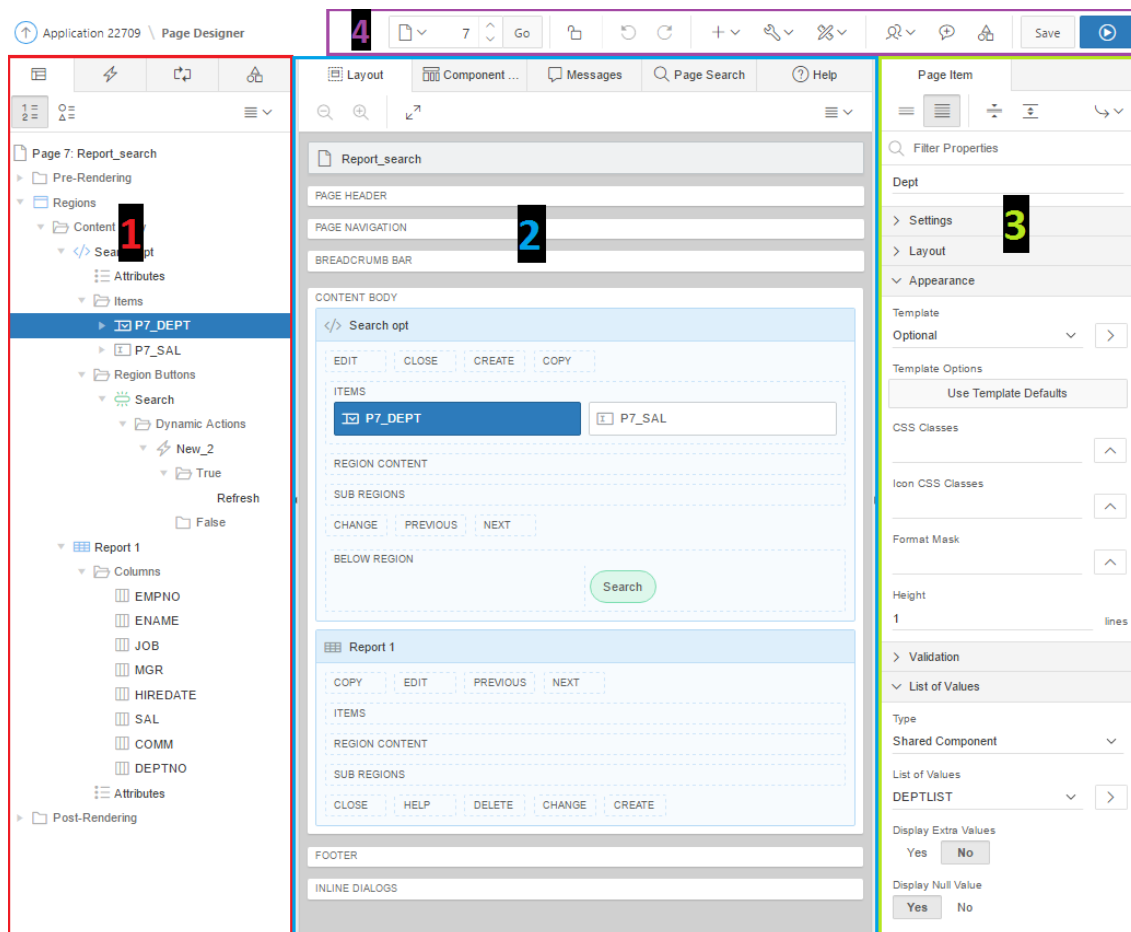


Obr. 4 ikona Application builder

Hlavní modul, kde se dají vytvářet a testovat webové aplikace (nebo stránky) v prohlížeči. Dále je zde možnost importu, či exportu aplikací a jejich přehled s možností prohlížení.



Co se týče způsobu tvorby, tak může trochu připomínat redakční systém, protože probíhá zajímavým systémem přednastavených šablon. Tedy vybere se určité schéma, což je kolekce šablon, která definuje uživatelský interface a dá mu určitý vzhled. Ten je možno dále upravovat a pro každou stránku navolit jednotlivé komponenty (např. tlačítka), které mají již také své šablony, ke kterým se vztahují široké individuální možnosti nastavení jednotlivých druhů komponent (např. umístění, název, jejich vzhled,... atd.). To vše se dá většinou provádět bez jakéhokoliv zápisu kódu, pouze přes různé nabídky, textová pole, radio buttony, atd. (tedy klasické kontrolky prostředí Windows). Všechny tyto možnosti nabízí nástroj zvaný **Page designer**, který je k dispozici u každé stránky aplikace. [10]

Struktura tohoto nástroje je podrobněji popsána dle Obr. 5.

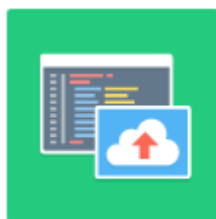


Obr. 5 rozložení nástroje Page designer

1. **Levý panel** – stromová struktura, která obsahuje komponenty na dané stránce
2. **Centrální panel** – představuje vizuální reprezentaci stránky. Je zde tedy možné si rozvrhnout jednotlivé komponenty dle potřeby a to stylem drag & drop.
3. **Pravý panel** – slouží pro nastavení vlastností daných komponent a dle toho je také nazýván jako property editor (neboli editor vlastností)

4. **Toolbar** – obsahuje sadu tlačítek, kde se dá například přes Redo change  a Undo change  pohybovat o krok dopředu a zpět při provedení nějaké změny, dále je zde užitečná možnost pro mazání stránek, která se nachází v rolovacím menu pod ikonkou klíče a nezbytná jsou zde také tlačítka pro uložení, spuštění, nebo volbu stránky, se kterou chce vývojář pracovat. [10] [2]

### 3.4.2 SQL Workshop



Obr. 6 ikona SQL Workshop

Jak již z názvu (v překladu SQL dílna) plyne, tak zde se provádí SQL dotazy nad databází, mění její strukturu, či obsah. Jak již bylo řečeno v úvodu o modulech, tak tento typ není k tvorbě podmínkou, ale ulehčí vývoj. Možno si pak díky němu připomenout, nebo modifikovat strukturu databáze, nebo prozkoumávat a upravovat její obsah. Také se zde nabízí ladit SQL dotazy, nebo PL/SQL bloky. Mezi nejdůležitější nástroje zde patří:

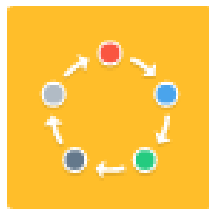
- **Object browser** – nástroj dobře uzpůsobený k průzkumu databáze, který obsahuje klasické činnosti, bez nutnosti psaní SQL, jen díky prostředí typu „point and click“ (je tedy vhodný i pro nezkušené vývojáře). Například jsou zde možné tímto způsobem akce pro tvoření, úpravu, či mazání tabulek, sloupců, nebo i sekvencí. Lze zde nalézt i další funkce, kterými může být třeba tvorba triggerů<sup>7</sup>, či procedur, ale zde se už bez znalosti PL/SQL vývojář neobejde.

---

<sup>7</sup> Trigger, se používá jako jakási spoušť k definici činnosti, která se má vykonat v případě práce nad tabulkou (např. při úpravě, nebo smazání)

- **SQL commands** – jedná se o konzoli, do které programátor zapisuje SQL dotazy, které jsou postupně po jednom vykonávány nad databází. Protože se zde provádí pouze čistý SQL zápis, tak je konzole vhodná zejména pro zkušené vývojáře. Poté se může stát i mocnějším nástrojem, než je výše uvedený Object browser. [10]

### 3.4.3 Team Development

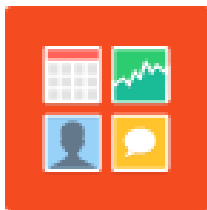


Obr. 7 ikona Team Development

Modul podpory týmové práce více členů, na společném projektu. Díky němu není nutno mít instalovaný v počítači software se stejnými funkcemi od jiného tvůrce, čímž tento nástroj usnadní a urychlí práci, během sdílení projektu s ostatními tvůrci. Není však nutnost toto využívat jen při práci v týmu, protože mnoho věcí může vývojáři sloužit jako přehled průběhu vývoje.

Je zde tedy možné si například stanovovat cíle, zjišťovat, zda vše běží podle předem stanoveného plánu (na to slouží nástroj **Milestones**), dále pokud někdo z týmu narazí na chybu, tak je možné jí pomůckou zvanou **Bugs** nahlásit, čímž se ostatní členové dozvědí o její přítomnosti a mohou jí nějakým způsobem řešit. Dále je tu **Feedback**, který je považován za nejdůležitější nástroj z tohoto modulu. Ten poskytuje mechanismus komunikace s koncovými uživateli, vývojáři a obchodními analytiky. V Team Development se také lze zabývat samotnými funkcemi přes položku s názvem **Features**. Zde je možné například popsat určitou funkcionalitu, která by se tam mohla přidat, nebo nějakou již existující změnit. Poslední možností je **To Do's**, kam se zaznamenávají jednotlivé části práce, které jsou potřeba vyřešit, případně jejich průběh. [9] [22]

### 3.4.4 Packaged Apps



Obr. 8 ikona Packaged Applications

Zde je možno do našeho prostředí instalovat doplňkové aplikace, které jsou plně podporované společností Oracle (např. aplikace Bug Tracking umožňující hledat chyby v kódu naší webové aplikace), nebo si nainstalovat tzv. Sample Code, který demonstruje určitou techniku, či koncept, což si lze dle potřeby přizpůsobit a využít ve své aplikaci, či webové stránce (např. Sample File Upload and Download), kde se dá dozvědět, jak zajistit načítání pomocí dialogů, či stahování souborů přes BLOB sloupce v databázových tabulkách. [9]



## 4 Tvorba výukové aplikace

Tato část práce se bude zabývat tvorbou a popisem výukové aplikace pomocí vývojového prostředí APEX.

### 4.1 Rozbor aplikačních modulů

Na základě přijatých podkladů týkajících se věcí, které jsou pro studenty problematické během vývoje aplikace v APEX, budou následující kapitoly tvořeny z modulů obsahující vybraná témata.

První modul pojednává o založení nové aplikace. Tedy cílem je, aby se čtenář dokázal zorientovat v počátcích jejího vzniku a znal možnosti, které v této fázi APEX nabízí. Druhý modul ve stručnosti představí datový typ BLOB, díky kterému jdou do databáze ukládat obrázky a nechat si je tak například zobrazovat v tabulkách. Následující téma řeší problematiku týkající se vzhledu a stylování celé aplikace. Čtvrtý modul bude zaměřen na vkládání dat do více tabulek z jednoho formuláře, tedy aby byl vývojář schopný provést tuto akci nad množinou dat, která se k sobě vážou, ale jsou rozdělena do více tabulek (např. zákazník a jeho adresa bydliště). O bezpečnosti aplikace z hlediska možností ověřování uživatelů a řízení jejich přístupu zavedením jejich rolí v systému pojednává modul č. 5. Předposlední téma objasní důležitý pojem session, který hraje klíčovou roli v možnosti zavést do aplikace, například v e-shopech, klíčový nákupní košík. Poslední modul navazuje na problematiku internetových obchodů a nabídne možnost přidat do sloupce tabulky klikatelná tlačítka, či odkazy, čímž se třeba může zpřístupnit možnost zobrazit detail položky produktu.

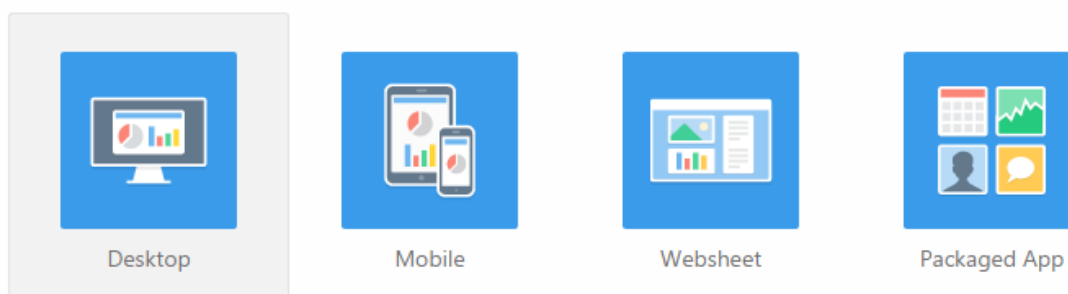
Paralelně s touto prací vznikl výukový program, který lze nalézt pod tímto odkazem: <https://ora1.uhk.cz/apex/f?p=114:1>. V něm se nachází pod jednotlivými záložkami (číslováním a problematikou shodná s moduly zde) výuková videa s popisným textem.

### 4.1.1 Modul 1. – Založení nové aplikace

V počátcích vývoje je nutno nejprve projít procesem, ve kterém vývojář zakládá novou aplikaci, přizpůsobenou dle jeho vlastních potřeb, tedy například správně navolená platforma pro budoucí program, způsob kterým jí bude chtít vytvářet, nebo zde bude moci přednastavit časové formáty a ověřovací schémata, která souvisí se zabezpečením aplikace.

#### Typ zakládané aplikace

Nový program může být typu pro **desktop**, kterým se myslí přizpůsobení obrazovkám zařízení jako je laptop, klasický stolní počítač, případně tablet. APEX nabízí ještě alternativu vývoje aplikace pro **mobilní zařízení**, aby se uživatelský interface přizpůsobil malým přístrojům typu smartphone a také zahrnuje jejich speciální úlohy, což může být například překlopení obrazovky či dotykové funkce. Co mají ovšem oba druhy (Desktop a Mobile) společného, je plný přístup k vývojovému procesu. Vývojáři tedy i umožní zapojit jeho programovací schopnosti a znalosti jazyka PL/SQL a SQL.



Obr. 9 možné typy nové aplikace

APEX však nabízí ještě další dva typy založení aplikace, přičemž jedním jsou tzv. Packaged App (tedy Zabalené aplikace) a Websheet. Funkce, kterou má první případ s názvem **Packaged App** se shoduje s tou, která je uvedena v bodu 3.4.4 Packaged Apps, tedy je možné si do vývojového prostředí doinstalovat doplňkové aplikace podporované společností Oracle. Druhým z těchto typů a celkově posledním, je tvorba

tzv. **Websheet**. U něj není potřeba vývojáře, pro vytvoření rychlých webových stránek. Administraci webového obsahu a jeho struktury si řídí sám koncový uživatel. Jeho použití je vhodné při záměru co nejrychleji a pohodlně vystavit obsah na internetu. [23]

### Ověřovací schémata

V průběhu zakládání aplikace, se nastavují i tzv. ověřovací schémata. Vzhledem ke skutečnosti, že každý uživatel v aplikaci má různá práva, se musí přihlásit do vytvořené aplikace, kde je potřeba ověřit správnost jeho přihlašovacích údajů. K tomuto účelu slouží funkce, která se ovšem musí specifikovat v tzv. ověřovacím schématu. Metody jsou během procesu vytváření aplikace celkem tři:

- **Bez ověřování (No Authentication)** – pomocí jazyka PL/SQL se nakonfiguruje tzv. DAD (Database Access Descriptor), což je jakýsi popisovač, kde jsou uložena jednotlivá uživatelská jména a hesla. Pokud ale nebude definován, tak návštěvník aplikace bude automaticky přihlášen jako veřejný uživatel (tedy bez nutnosti zadání hesla), což je nejčastější využití tohoto schématu.
- **databázový účet (Database Account)** - k ověření uživatele se používá uživatelský účet, který je zapsán v lokální databázi. Díky tomuto typu lze dle vlastních potřeb spravovat účty pomocí databázových nástrojů (např. SQL\*Plus, se kterým se pracuje v bodě 3.2.2, týkající se instalace APEX). Jako test funkčnosti tohoto způsobu ověřování, se může vývojář později zkusit přihlásit do aplikace pomocí účtu instance administrátora, který mu bude jakožto databázový účet přístupný, případně lze v SQL\*Plus založit nového uživatele příkazem **create user [login] identified by [heslo]** (platí samozřejmě pouze v případě lokální instalace Oracle database, nad kterou běží instance APEXu).
- **účty APEXu (Application Express Accounts)** - jedná se o způsob ověřování pomocí údajů účtů, které již existují a to pro přihlašování do pracovního prostoru APEXu (tedy všechny účty, jak nové aplikace, tak i nadřazeného workspace budou sjednoceny). [24]

## Taby

Během průvodce založení nového programu lze narazit na pojem **Tabs**. Jedná se o nastavení navigace v aplikaci. Tato položka ovšem nebude dostupná, pokud byla v jednom z předchozích kroků wizardu zachována šablona s názvem Universal theme, která již využívá pro navigaci jiný způsob (o Themes více v kapitole 4.1.3).

Při práci s taby, se vyskytuje několik pojmů – tedy samotný **Tab**, který představuje navigační položku, **Tab set**, což je celá sestava těchto prvků a v případě volby složitější navigace, se pracuje i s pojmem **Parent tab**, jakožto nadřazenou záložkou pro každý tab set. V průvodci tedy budou na výběr tyto možnosti:

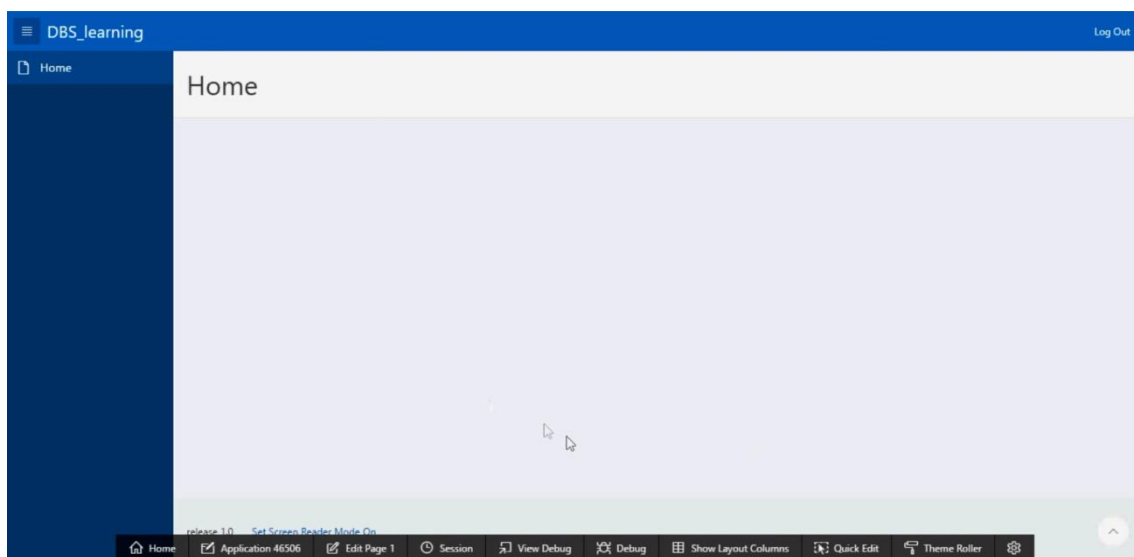
- **No Tabs (bez tabů)** – pro pohyb v aplikaci se budou používat jiné prvky (např. tlačítka, seznamy, atd.), které nabízí webová šablona.
- **One-Level Tabs (jednourovňové taby)** - Navigace bude tvořena pouze z jednoho tab setu (odpadají parent taby)
- **Two-Level Tabs (dvourovňové taby)** – lze vytvářet kromě tabů i tabsety s parent taby, což se provádí přes **nazev\_aplikace > Shared components > Tabs**  
[25]

## Časové formáty

V databázích se často vyskytují informace týkající se data a případně i času (např. kdy byla položka vytvořena, upravena, nebo v případě obchodních listin jako jsou objednávky, nebo faktury, tak i datum vyřízení). Proto by mohlo být užitečné si již při zakládání aplikace stanovit, jakou formou se budou zobrazovat různé takové informace v datových typech napříč aplikací. Celkově se rozlišují 4 druhy, nesoucí tento typ formátu:

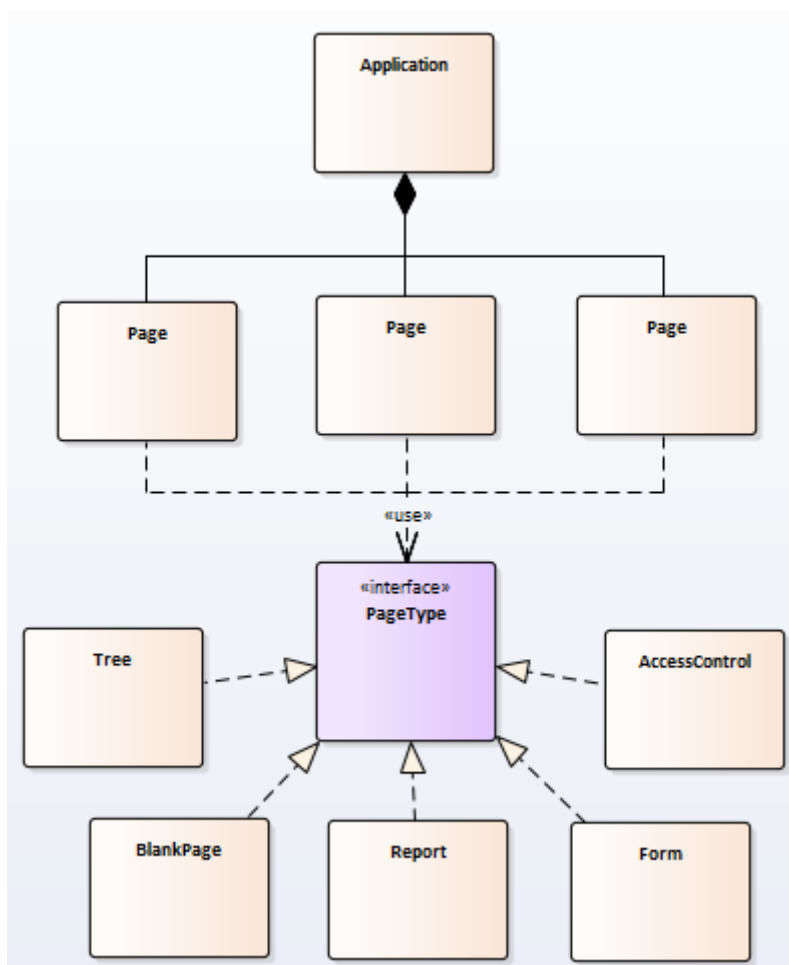
- **Formát data (Date Format)** – určuje, jakým způsobem budou proměnné datového typu DATE v aplikaci zobrazovat datum (tedy zvolí se formát pro jeho zobrazení). Příkladem může DD-MON-YYYY pro den-měsíc-rok.

- **Formát data času (Date Time Format)** – volí se formát pro prvky, které v sobě kromě data nesou i časovou informaci (např. DD-MON-YYYY HH:MI:SS, což znamená den-měsíc-rok hodina:minuta:sekunda)
- **Formát časové značky (Timestamp Format)** – stanovuje se formát proměnných nesoucí informaci vzniku nějaké události.
- **Formát časové značky u časové zóny (Timestamp Time Zone Format)** – určení formátu zobrazení hodnot, které navíc obsahují kromě informace vzniku nějaké události i data o časovém pásmu [26]



Obr. 10 ukázka pohledu na nově vytvořenou aplikaci

## Obsah nově vytvořené aplikace



Obr. 11 UML class diagram znázorňující vztahy mezi aplikací, stranami a jejími typy  
Zdroj: vlastní tvorba

Aplikace jsou v základu skupiny stránek, které provádí daný úkol (nebo jejich množinu), vztahující se k business funkci. Tedy vytvoření nového programu pomocí wizardu, je vlastně jen definování takřka prázdného těla (v základu je vytvořena pouze prázdná Home strana čekající na obsah a přihlašovací formulář). Pokud by si tedy vývojář přál vytvořit další webovou stránku, tak je nutno určit její typ. Ten může být:

- **Blank page (Prázdná stránka)** – neobsahuje žádnou vestavěnou funkcionalitu. Vývojář má tedy kompletní kontrolu nad budoucím obsahem, který může být složen např. jen z čistého HTML a případně až později doplnit nějakou z nabízených funkcí APEXu (např. report, form, atd.).
- **Report** – jedná se o tabulkový výpis realizovaný pomocí SQL dotazu. Některé možnosti se s níže uvedeným formulářem slučují (např. možnost reportu nad formulářem).
- **Form (Formulář)** – slouží jako uživatelský interface pro vykonání SQL dotazu nad tabulkou, přesněji pro manipulaci s jejími řádky.
- **Calendar (Kalendář)** – vytvoří kalendář obsahující události na určité období (dny, týdny, nebo měsíce). Je nutno mít připravenou tabulku, ze které si tato komponenta vezme potřebné údaje (především počáteční a koncové datum události spolu s jejím popisem). Na každé stránce může být pouze jeden kalendář.
- **Acess control (Řízení vstupů)** – viz bod 4.1.4
- **Feedback** – přidá stránku, která se vztahuje k modulu s názvem Team development. S její pomocí lze sbírat od uživatelů aplikace zpětnou vazbu ve formě komentářů. Tato stránka bývá v podobě pop-up okna<sup>8</sup>, které je možno aktivovat např. zasazením do globální navigace aplikace (tedy tam, kde se dá najít tlačítko pro přihlášení/odhlášení uživatele). Odtud je pak možno vkládat zpětnou vazbu pro každou zobrazenou stránku, která se uživatelům s příslušnými pravomocemi zobrazí v APEXu pod cestou **Team development > Feedback**.
- **Tree** – slouží k hierarchickému zobrazení dat v tabulkách, nebo views. Příkladem může být tabulka zaměstnanců, která obsahuje sloupce s ID zaměstnance a zároveň ID manažera. Pomocí wizardu a těchto dvou klíčových sloupců se namapují data a následně je možno vidět graficky jejich hierarchii.

---

<sup>8</sup> Tzv. „vyskakovací“ okna, která se mohou objevit na určité webové stránce. V případě aktivace pop-up element se objeví před stranou (tedy překryje obsah), díky které se aktivoval. Často se objevují formou reklam. Bývají rušivá a nežádoucí.

- **Wizard (Průvodce)** – vytvoří klasického průvodce
- **Data Loading (Načítání dat)** - jedná se o zavedení dalšího wizardu, se zaměřením na jednoduché načítání dat do tabulek (z CSV<sup>9</sup> souborů, nebo prostým zkopírováním textu). Vzhledem k tomu, že jsou zapsány do databáze, tak je lze poté libovolně sdílet přes schéma, které je této aplikaci přidruženo.
- **Chart (Graf)** – možnost vytvoření grafu na základě výběru klíčových sloupců pro hodnoty z tabulek, nebo pohledů
- **Plug-ins (zásuvné moduly)** – do aplikace lze přidat rozšíření, které se spravuje podobně, jako dodatečná komponenta. Jde o různé typy prvků (např. regiony<sup>10</sup>, dynamické akce<sup>11</sup>, procesy<sup>12</sup>, atd.), které vyprodukoval někdo jiný. Příkladem může být region plugin, přidávající na stránku chat. Aby bylo možno přidat tento modul, tak je nutno nejprve nějaký nainstalovat. To lze provést pod cestou **nazev\_aplikace > Shared components > Plug-ins**.
- **Global page** – vytvoří jednu speciální stránku, která se bude zobrazovat na všech ostatních v aplikaci (tedy pokud bude například obsahovat malou ikonku s logem, tak každá stránka v programu jí bude zobrazovat též) [27]

#### 4.1.2 Modul 2. – Práce s obrázky v tabulkách

Pokud by chtěl vývojář přidat do tabulky možnost stahování či zobrazování obrázků (případně dokumentů, videí,... atd.), tak v takovém případě využije datový typ BLOB (Binary Large Objects). Jeho funkcí je ukládání binárních souborů ve sloupci (=deklarativní BLOB). Je možné navíc využít dalších atributů, který doplňují soubor o další informace:

---

<sup>9</sup> Souborový formát určený pro výměnu tabulkových dat.

<sup>10</sup> Region je prvek tvořící oblast na stránce, která slouží jako kontejner pro její obsah (může trochu připomínat v HTML tag <div />)

<sup>11</sup> Využívání dynamických akcí se snaží ulehčit práci vývojářům, kteří nedisponují znalostmi JavaScriptu. Jejich využití se tedy najde např. u validací formulářů, alertů (vyskakujících upozornění), zobrazování/skrývání elementů na základě podmínek, atd.

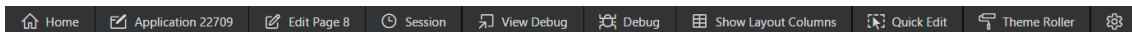
<sup>12</sup> Jde o definici procesů, které mohou nastat během určité činnosti, např. co se stane po stisknutí tlačítka pro odeslání formuláře.



- FILENAME – obsahuje název souboru, který byl nahrán
- MIME\_TYPE – uloží typ souboru, takže s ním prohlížeč bude moci podle toho pracovat
- LAST\_UPDATED – obsahuje datum poslední aktualizace daného BLOBu [2]

### 4.1.3 Modul 3. – Stylizace šablony

Problematika týkající se vzhledu a stylování webových stránek v prostředí APEX má kořeny v tzv. **Themes**, což jsou kolekce šablon, které stanovují layout (rozložení stránky) a vzhled celé aplikace. Základní theme se nastavuje již při jejím vytváření. Pro stylování stránek je uzpůsobeno nejnovější theme s názvem **Universal theme**. Tento typ šablony se dá rychle a pohodlně upravovat pomocí dvou tlačítek, která jsou dostupná z toolbaru v testovacím módu aplikace – **Quick edit** a **Theme Roller**.




Obr. 12 ukázka toolbaru v testovacím módu aplikace

#### Quick edit

Funkce tohoto tlačítka spočívá v tom, že si vývojář najede kurzorem myši na nějaký prvek na stránce (např. tlačítko, textové pole, region,... atd.) a následně má dvě možnosti – po zvolení příslušného elementu (mimo ikonku klíče v pravém horním rohu ohraničení) se zpřístupní jeho nastavení v Page designeru, kde lze provádět úpravy.

Sample Database Application - Customer Details

\* First Name  

\* Last Name

Street Address  Line 2

City  \* State

\* Zip Code

\* Credit Limit

Phone Number  Alternate Number

Email  URL

Tags

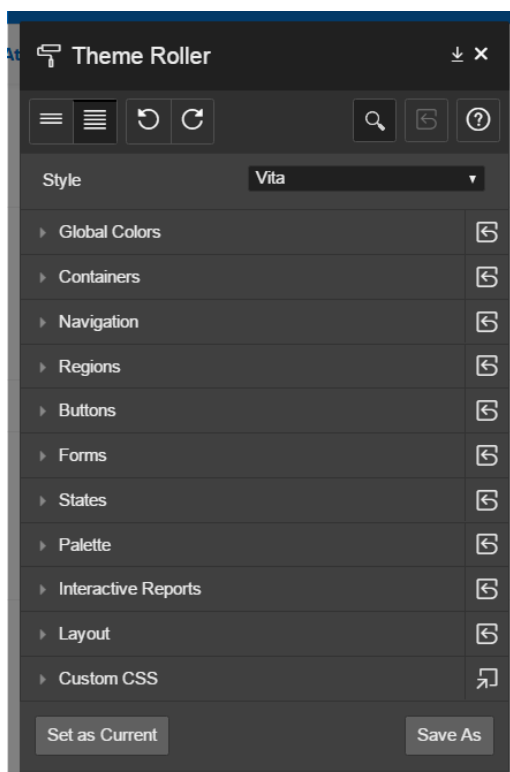
Cancel

Obr. 13 ukázka funkce Quick edit

Zdroj: [3]

Jako druhá možnost je poklepnání na ikonku s obrázkem klíče, přičemž se zpřístupní tzv. **Edit Live Template Options**, neboli Editace nastavení šablony živě. Přes tento nástroj lze za běhu aplikace měnit například velikost, celkovou stylizaci prvku, odsazení a další zajímavé věci. Do tohoto menu se však dá dostat i mimo testovací mód z Page designeru, po výběru daného prvku, v pravém panelu podmenu s názvem **Appearance**, kde se nachází popisek **Template options** a pod ním tlačítko. [28]

## Theme roller



**Obr. 14** ukázka pohledu na nově vytvořenou aplikaci

Dle Obr. 14 je vidět ona pomůcka, která si poradí především s barevnou stylizací specifických elementů. Ovšem dají se přes ni i zaoblit u elementů hrany, stanovit poměry mezi prvky na stránce a případně přes poslední možnost s názvem **Custom CSS** je možno vkládat libovolný CSS kód, který je okamžitě prováděn a následně bude přidán na konec kaskádových stylů. Po provedení potřebných změn na stránce, se dá přes tlačítko s názvem **Save As** uložit nový styl, případně vyresetovat ikonkou se šipkou v pravé horní části okna. V nové verzi APEX byla přidána i další možnost **Set as Current**, která způsobí, že daný styl bude nastaven jako standartní (tedy začne se ihned využívat právě onen, což v minulosti nebylo takto pohodlně možné). [29]

#### 4.1.4 Modul 4. – Vkládání dat do více tabulek z jednoho formuláře

Často se může vývojář setkat s problémem, kdy bude po své aplikaci chtít, aby se pracovalo s jedním formulářem nad více tabulkami. Příkladem může být vývoj aplikace sloužící jako chat. Tam budou tabulky, kde jedna bude obsahovat uživatele a další zprávy od nich. V rámci jednoho záznamu bude tedy nutno provést najednou zápis do několika tabulek, které na sobě budou závislé. K tomu bude nejvhodnější konstrukt zvaný procedura.

##### Procedura

Lze je v APEXu najít souhrnně v modulu **SQL Workshop > Object browser**, kde se dají celkově spravovat (možno samozřejmě využít i konzoli - SQL commands). Procedura je považována za subprogram (podprogram), co provádí danou akci na datech v databázi. Tím se myslí pojmenovaný blok PL/SQL kódu, který lze díky tomu, že je pojmenovaný, znovu po uložení z databáze vyvolat a použít (např. v triggeru, jiné proceduře, anonymním bloku – tedy nepojmenovaném, atd.). Z toho plyne výhoda, která zabraňuje duplicitám v kódu (tedy při změně kousku kódu stačí provést změnu pouze na jednom místě). Výkon kódu v těle se provádí atomicky - buďto se provede všechno, nebo nic (odpadá nekonzistence dat). [12]

Struktura procedury:

```
CREATE [OR REPLACE] PROCEDURE nazev_procedury (seznam parametrů)
```

```
IS
```

```
    Deklarace proměnných
```

```
BEGIN
```

```
    Výkon daných příkazů
```

```
END;
```

Jak je vidět z předchozí konstrukce, tak procedura se skládá z názvu, deklarační a výkonné části, ale navíc za názvem může obsahovat parametry (nula, jeden, nebo více), čímž se stávají mocným nástrojem. Jednu proceduru lze totiž velmi zobecnit a využívat v mnoha situacích stejně jako například ve skutečném životě šrouby. Proto jsou jejich parametry v praxi využívány ve většině případů. Celkem existují tři typy:

- **IN (vstupní)** – Je pouze read-only, tedy pro čtení - nelze ho uvnitř procedury měnit. Pokud se vůbec typ parametru při zakládání procedury nedefinuje, tak je implicitně nastavený jako základní.
- **OUT (výstupní)** – Je určen pouze pro zápis (nelze z něj číst)
- **IN OUT (vstupně-výstupní)** – vstupně výstupní. Tedy proměnná je připravena ke čtení a následné úpravě a vrácení.

Pro lepší představu funkčnosti parametrů v praxi je níže uveden příklad, který zahrnuje použití všech tří typů. Jedná se o proceduru, co analyzuje zaměstnance vzhledem k jejich pozici ve firmě (očekávaná výstupní OUT proměnná), dle jejich ID (které se načítají jako IN parametr). Zároveň je implementováno počítadlo, určující, kolik již bylo vyhodnoceno zaměstnanců (protože se očekává, že již obsahuje nějakou hodnotu, která bude následně modifikována a vrácena. Proto představuje vstupně výstupní IN OUT parametr). [30]

#### **Vytvoření procedury:**

```
CREATE OR REPLACE PROCEDURE get_job_of_someone (empno_in IN NUMBER, job_out OUT
VARCHAR2, trial_count_in_out IN OUT NUMBER)
IS
BEGIN
    select job into job_out from emp where empno = empno_in;
    trial_count_in_out := trial_count_in_out + 1;
END;
```

### Otestování přes anonymní blok:

```
DECLARE
    job_l VARCHAR2(40);
    empno1_l NUMBER(6,0):= 7698;
    empno2_l NUMBER(6,0):= 7839;
    trial_count_l NUMBER := 0;
BEGIN
    get_job_of_someone(empno1_l, job_l, trial_count_l);
    DBMS_OUTPUT.put_line('Zaměstnanec s ID ' || empno1_l || ' vykonává zaměstnání ' || job_l
    || '. Celkový počet zanalyzovaných zaměstnanců: ' || trial_count_l);
    get_job_of_someone(empno2_l, job_l, trial_count_l);
    DBMS_OUTPUT.put_line('Zaměstnanec s ID ' || empno2_l || ' vykonává zaměstnání ' || job_l
    || '. Celkový počet zanalyzovaných zaměstnanců: ' || trial_count_l);
END;
```

### Výstup:

Zaměstnanec s ID 7698 vykonává zaměstnání MANAGER. Celkový počet zanalyzovaných zaměstnanců: 1  
Zaměstnanec s ID 7839 vykonává zaměstnání PRESIDENT. Celkový počet zanalyzovaných zaměstnanců: 2

### Formulář nad procedurou

Po objasnění pojmu procedura se lze vrátit k problematice, týkající se vkládání dat do více tabulek. V prostředí Apex, při zakládání nové stránky, je k dispozici typ s názvem **Form on a procedure**, plus samotná procedura, která v něm bude zapojená a nejlépe ještě komponenta typu **Report** pro zobrazení výsledných dat z obou tabulek.

Jak už z názvu plyne, jedná se o formulář, který bude přijímat uživatelská data a bude vytvořen na základě vybrané procedury.



Obr. 15 ikona Form on a procedure

Pro jeho vytvoření je nejjednodušší možnost využít wizard (průvodce), který provede vývojáře celkem třemi fázemi. Nutno zmínit, že výsledný formulář bude v základu obsahovat dvě tlačítka - Cancel a Submit. Oběma je nutno nastavit stránku, na kterou uživatele po stisknutí přesměrují (položky **Branch Here on Submit a Cancel and Go To Page**). Užitečný by mohl být i **Page mode**. Jedná se o mód stránky, kde jsou na výběr dvě možnosti:

- **Modal Dialog** – okno, které se vytvoří určitou akcí (např. vytvoření nové zprávy pro chat). To je poté svázané s komponentou pod ním (např. se stránkou, kde se nachází výpis tabulky se zprávami od uživatelů). Než bude moct uživatel pracovat se stránkou v pozadí, ke které je dialog svázán, tak ho nejprve musí nějakým způsobem ukončit (tím může být vytvořením nové zprávy do chatu, případně ukončení dialogu křížkem).
- **Normal** – vytvoří formulář, který bude obsažen na klasické stránce

V posledním kroku, po výběru procedury, navíc APEX sám navrhne dle její parametrů formulářové komponenty a jejich názvy. [31]

#### **4.1.5 Modul 5. – Security (bezpečnost)**

U větších webových stránek bývá dost nepravděpodobné, aby měl každý uživatel přístup k veškerému obsahu a možnostem dané aplikace. V takovém případě je nutno jednotlivce ověřit a stanovit jim práva v systému.

##### **Ověřovací schémata**

To, jakým bude prováděno ověřování, se stanovuje na základě ověřovacích schémat. Ta se spravují přes **Shared components > Authentication Schemes**. Při jejich volbě je dobré vědět, že v aplikaci může být více ověřovacích schémat, ale platné je v danou dobu vždy jen jedno. Po výběru ověřovacího schématu je vhodné provést i nastavení samotné aplikace a to pro každou stranu, případně i pro její záložku v navigaci (pokud se pomocí ní zobrazuje). Nezákladnější ochranou je možno po uživateli vyžadovat alespoň přihlášení. To se dá základně nastavit přes nástroj web designer, ve skupině Security a položky Authentication. Pokud by chtěl vývojář využít složitější systém, kde

nebude záležet jen na podmínce registrace v systému, ale i rolí jednotlivých uživatelů, tak by mohl sáhnout po pomůcce s názvem **Access control**.

Způsoby ověřování uživatelů byly částečně popsány v modulu 4.1.1, během zakládání nové aplikace. Tam šlo vybrat jedno ze tří schémat - Bez ověřování, Databázový účet a Účet APEXu. Za zmínku stojí ještě další dvě a to:

- **LDAP<sup>13</sup> Directory scheme (schéma LDAP adresáře)** – uživatelský účet je spravován pomocí adresáře LDAP, který vlastní každý zaměstnanec ve firmě. Z důvodu jednoduchosti administrace je velmi využíván v mnoha organizacích.
- **Custom scheme (vlastní schéma)** – pokud vývojáři nevyhovuje ani jeden z předchozích typů ověřování, tak je možno vytvořit vlastní způsob autentizace (důvodem může být např. vlastní návrh přihlašovacího formuláře). Během zakládání tohoto typu, se v průvodci objeví pole celkem pro čtyři funkce, ale obvykle se využívá jen **Authentication Function Name**, kde se očekává (pokud byla vývojářem v minulosti založena) její název. Tedy pokud neexistuje, lze jí vložit do dolního pole pro PL/SQL kód. Tato funkce má však pravidla – musí obsahovat návratovou hodnotu boolean (vrací true/false, což se dá vyložit jako „úspěšně přihlášen“, nebo „špatně zadané údaje“) a přijímat alespoň dva základní parametry pro uživatelské jméno a heslo. To by ale nemělo být v tomto případě jako plain (prostý) text, ale šifrované pomocí nějaké hashovací funkce. Určitě by bylo navíc vhodné vytvořit v databázi tabulku s uživateli pro tuto funkci (sice bude platná i bez ní, pokud se v ní návratová hodnota např. nastaví vždy na true, ale pak by ověřování postrádalo smysl). [10]

---

<sup>13</sup> LDAP (Lightweight Directory Access Protocol) - Adresářová informační služba. Obsahuje a spravuje informace o dané věci, nebo osobě (případně informace o osobním nastavení)



Authentication Schemes			
Name ↑	Scheme Type	Subscribed From	Subscribers
Application Express Authentication - Current	Application Express Accounts		

Obr. 16 ukázka tabulky autentizačních schémat

### Access control

Access control je druh stránky v aplikaci, která slouží jako řízení uživatelských přístupů speciálním seznamem (access control list) a jeho definici. Po jeho založení mohou mít účty jednu ze tří základních rolí autorizačního schématu a to:

- **view** - pro uživatele, kteří si mohou jen prohlížet obsah
- **edit** - může prohlížet a upravovat obsah webu
- **administration** – uživatel s veškerými právy

Application Administration

Application Mode

- Full access to all, access control list not used.
- Restricted access. Only users defined in the access control list are allowed.**
- Public read only. Edit and administrative privileges controlled by access control list.
- Administrative access only.

---

Access Control List

Identify usernames which correspond to this application's authentication scheme.

Find

<input type="checkbox"/>	Username ↑	Privilege	Last Changed By	Date
<input type="checkbox"/>	PAVEL	Edit	relycan1	69 minutes ago
<input type="checkbox"/>	JANA	View	relycan1	69 minutes ago
<input type="checkbox"/>	DAVID	Administrator	relycan1	112 minutes ago

1 - 3

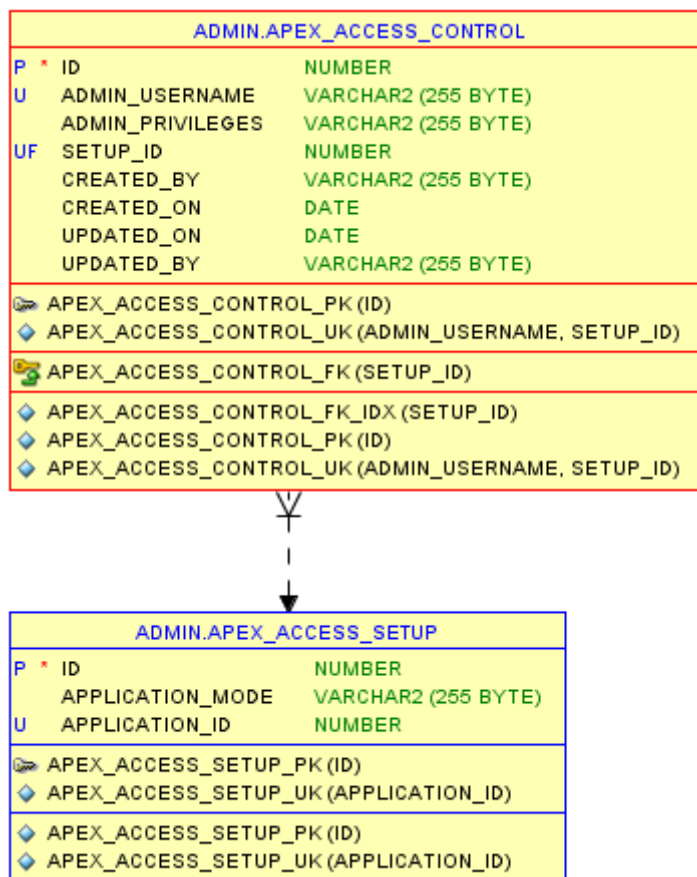
Obr. 17 ukázka stránky s řízením uživatelských přístupů

Na samotné stránce access control je však možné (a často nutné) nastavit aplikační mód, který může fungovat celkem čtyřmi způsoby, kde se mohou upravit pravomoce tohoto schématu:

- **Full access to all, access control list not used** – základní nastavení access control listu (ACL), na které je dobré si dát pozor, protože všechny role budou mít veškerá práva (access control se tedy v tomto případě nebude vůbec používat!)
- **Restricted access. Only users defined in the access control list are allowed** - pouze uživatelé, kteří jsou obsaženi v access control listu s přidělenou rolí mohou prohlížet obsah (tedy ani pravomoce pro roli view nejsou těmto rolím dostupné)
- **Public read only. Edit and administrative privileges controlled by access control list** – asi nejpoužívanější způsob, kdy všichni uživatelé, kteří nejsou v access control listu mají stejné pravomoce jako role view. Ostatní funkce uživatelů se budou řídit klasickými podmínkami rolí autorizačního schématu
- **Administrative access only** - kromě role administrátora budou všechny ostatní pravomoce smět jen prohlížet obsah [32]

Po vytvoření Access control listu vzniknou navíc:

- Dvě navzájem propojené tabulky (viz obr. 18) – první nese název APEX\_ACCESS\_CONTROL. Ta obsahuje názvy uživatelů (sloupec ADMIN\_USERNAME), jejich přidělené role (ADMIN\_PRIVILEGES) a cizí klíč (SETUP\_ID), kterým se váže k druhé tabulce APEX\_ACCESS\_SETUP. V té je určený aplikační mód (APPLICATION\_MODE), spolu s identifikátorem aplikace (APPLICATION\_ID), ke které se Access control vztahuje.



Obr. 18 ukázka modelu pro Access control

- autorizační schémata<sup>14</sup>, která se vztahují k aplikačnímu módu Access control listu (vytvoří je pod názvem access control – administrator, access control - edit, access control – view)
- privilegia v něm obsažená (tedy v základu Edit, View, Administrator)

Na závěr by bylo užitečné zmínit, že Access control trpí nedostatkem a to omezením pouze na tři přednastavené role (viewer, editor a administrator). V takovém případě by bylo nutno si napsat vlastní skript pro řízení uživatelských přístupů. [2]

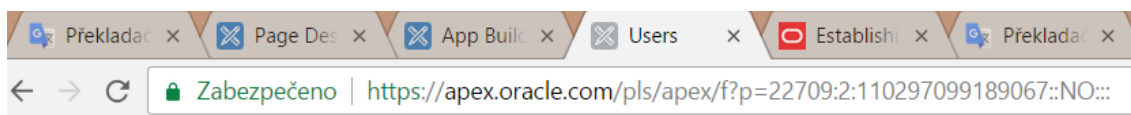
---

<sup>14</sup> Autorizační schémata definují z první poloviny to, jak uživatel vystupuje v aplikaci - jaká má práva (přidělí se mu role). Následně se toto schéma ještě použije do podmínky v dané komponentě aplikace, která se na tuto roli dotáže, případně se z důvodu nedostatečných práv uživateli nezobrazí (fungují tedy na způsob otázky i odpovědi)

#### 4.1.6 Modul 6. – Sessions

Během procházení webu je často užitečné, aby se uživateli udržela nějakým způsobem data, která se vztahují k jeho návštěvě (např. u e-shopů obsah košíku). V takovém případě se využije session (neboli relace), což je logický konstrukt, který slouží ke sdílení údajů mezi jednotlivými http požadavky a k identifikaci uživatele (servery si tedy díky sessions dokáží pamatovat jejich činnost). Vlastně tedy nejde o nic jiného, než schopnost pamatovat si stanovené hodnoty jednotlivých proměnných během procházení webových stránek.

APEX tento problém řeší tak, že ke každé session přidá identifikátor. Při spuštění aplikace ho lze nalézt v URL adrese, která obsahuje **f?p=**. Za znaménkem rovná se následují tři číselné řetězce oddělené dvojtečkami. [2] [9]



Obr. 19 ukázka struktury URL aplikace v APEX

Prvním je ID aplikace, druhý ID stránky a třetí není nic jiného, než právě ID session. Po odhlášení a novém přihlášení je interakci přidělen nový identifikátor.

Na hodnotu v session, která je uložena v textových polích a oblastech, heslech, select listech,... atd., se dá odkázat stylem **:NAZEV\_POLOZKY**. Tento název se dá dohledat například v Page designeru, v levém panelu, u elementů typu Item. Pokud se provede tento druh akce, tak je možno tímto způsobem zapisovat i do jiných komponent tohoto typu na jiných stránkách, případně na jednu a tu samou. Takovým příkladem může být prostý zápis **:POLOZKA1 := :POLOZKA2**. V session ovšem kromě proměnných lze zavést i virtuální tabulky, které je možné až později propsat do databáze. To je vhodné například právě pro nákupní košík, kde do něj uživatel přidá několik položek a teprve po potvrzení vystavení objednávky se fyzicky uloží. Tento způsob je k dispozici díky rozhraní s názvem **APEX\_COLLECTION**, které umožňuje tvořit kolekce.

Pro práci se sessions společnost Oracle připravila i pomůcku pro sledování aktuálních uložených proměnných. Dá se nalézt v testovacím módu aplikace, na toolbaru pod tlačítkem s obrázkem ciferníku a nápisem Session. Po rozkliknutí se v novém okně zobrazí informace (např. s ID aktuální session) a hlavně formulář, přes který se dají vyhledat a vypsat at' už uložené proměnné, nebo i všechny kolekce v session. [10]

Application ↑	Page	Item Name	Display	Item Value	Status	Encrypted
22709	20	VALUE1	Text Field	some text	Updated	No
22709	20	VALUE2	Text Field	some value	Updated	No

Obr. 20 ukázka výpisu stavu session

#### 4.1.7 Modul 7. – Vložení tlačítka do řádku tabulky

Někdy může nastat situace, že vývojář tvoří e-shop, kde má výpis tabulky s jednotlivými produkty. Pokud v takovém případě navštíví tuto stránku zákazník a zaujme ho nějaký ze seznamu výrobků, tak by si určitě přál, aby po kliknutí na název mohl zjistit i podrobnosti o dané položce. Avšak tento problém se nemusí týkat jen položek zboží internetového obchodu, ale i například výpisu článků nějaké redakce, zobrazení nabídky či poptávky na nějakém fóru a dalších mnoha věcí.

K jednoduchému řešení dané problematiky bude zapotřebí tedy nějaká již vytvořená tabulka s položkami, dále nějaká stránka na které se z ní budou vypisovat jednotlivé hodnoty a poté ještě jedna, co bude po kliknutí na položku zobrazena. Avšak zde bude na výběr ze dvou možností – vytvořit **modální dialog** – tedy aby se informace zobrazovaly v menším okně a po přečtení, se mohlo okno uzavřít a pohodlně

pokračovat dál na dalších řádcích (tedy vhodné například pro e-shopy a jejich výpis položek), nebo se může obsah zobrazit do **klasické nové strany** (alternativní řešení např. pro výpis článků).

Po vytvoření stránek bude nutno se přesunout do nástroje Page designer, v levém panelu stromové struktury pod reportem (výpisem tabulky) vybrat nějaký prvek představující sloupec a jeho formátování. Tam je možno, v pravém panelu v podmenu s názvem **Identification** a položkou **Type**, vybrat typ zobrazení pro daný sloupec. Pro klasický výpis hodnoty v buňce se využívá **Plain text**, ale je možné sloupec přes položku **Hidden Column** úplně skrýt, nebo např. zobrazit obrázek sloupcem typu BLOB (pro význam pojmu viz modul 2). V případě vytvoření tlačítka, nebo odkazu se bude využívat typ **Link**. Při jeho zvolení se vyrobí další podmenu v podmínce se stejným názvem, kde se nachází dvě položky a to **Target** a **Link text**.

Druhý z uvedených pojmů umožní si zvolit vzhled údaje ve sloupci. Je možno, aby se zobrazoval jako odkaz (tedy podtržený a zamodralý), případně jako tlačítko, nebo ikonka. V případě grafického zobrazení je na výběr několik přednastavených obrázků, ale je možno použít i vlastní, stačí do řádku napsat klasické HTML pro vkládání obrázku, v následujícím příkladu se jedná o cestu k lokálnímu adresáři daného workspace: ****

Po zvolení vzhledu odkazu, či tlačítka se vývojář přesune k položce **Target**, kde se po kliknutí na tlačítko otevře okno s názvem **Link Builder – Target**. Zde se zvolí, která stránka se má po přejití na odkaz zobrazit, ale také je důležité zmínit, že kromě toho se tu i přes položku **Set Items** dají nastavit hodnoty proměnných, které se promítnou v položkách na stránce, která bude po kliknutí zobrazena. To je výhodné například právě v situaci, kdy by se měly zobrazit detailní informace produktu. V takovém případě je za vhodné takové sloupce v tabulce skrýt a zobrazit je až ve výpisu. [33]

## 4.1.8 Závěrem o výukové aplikaci

Z předcházejících modulů je složena aplikace, která je uložena na školním serveru UHK a bude využita pro výukové účely předmětu Databázové systémy.



Obr. 21 ukázka výukové aplikace

Tento program se skládá ze stránek, obsahující videa uložená mezi tagy typu <video/>, jež jsou součástí HTML5. Mimo jich tu lze najít i logo umístěné na úvodní stránce. Tím je obrázek zobrazený v klasických <img/> tazích a uložený na serveru. Veškerý takový obsah i s naformátovaným textem obaluje kontejnerový prvek region. Následně je celá stránka vložena jako položka do navigace, kterou podporuje zde využitá kolekce šablon Universal theme (tedy nebyly použity taby jako tomu je u starších themes) a barevně upravená nástrojem Theme roller. Externě (také ze serveru jako logo) se i načítá CSS soubor, co aplikuje jednotným stylem, na každé stránce zarovnávání videí. Vzhledem k tomu, že obsah webu je veřejně přístupný a nevznikla potřeba přihlašovacího systému, nebo rolí, tak zde nebyl zaveden žádný způsob zabezpečení. Naopak je každá stránka nastavena jako veřejná.

## 5 Shrnutí výsledků

V první, teoretické části byl v první fázi objasněn pojem APEX. Kromě samotné definice se však navíc doplnily technologie, na kterých je postaven, nebo se kterými v něm může vývojář pracovat. Následoval popis zprovoznění vývojového prostředí, nabízející možnost lokální instalace, případně i bez ní. V prvním typu bylo navíc nutno nainstalovat i SŘBD s názvem Oracle Database, aby se dal APEX plnohodnotně využívat, zatímco způsob bez instalace odhalil stinnou stránku a to využití softwaru spíše na ozkoušení, případně ohodnocení programu (z důvodu omezených možností). Po instalaci následovala definice klíčových pojmů APEXu a rozbor jeho modulů.

V druhé části této práce, byla na základě stanovených podkladů vytvořena výuková aplikace, skládající se celkem ze sedmi modulů. Problematika každého z nich byla objasněna a na jejím základě k nim postupně vznikla videa s popisným textem.



## 6 Závěry a doporučení

Oracle APEX se už zpočátku jevil jako skvělým pomocníkem při rychlé tvorbě databázových aplikací. V této práci se čtenář o jeho kvalitách může přesvědčit zejména ve čtvrté kapitole, během průzkumu jednotlivých modulů, kde jinak poměrně složité věci (ukázkovým příkladem je například zabezpečení), je možno provést v pár krocích. Navíc jeho zprovoznění, pro možnost ozkoušení, zabere jen pár minut. Trochu problém by však mohl nastat, kdyby čtenář zatoužil po českých zdrojích informací, protože většina z nich je spíše v anglickém jazyce. Na internetu však lze nalézt mnoho video návodů a skvělých anglických knih s praktickými postupy (viz seznam použité literatury), takže i přes tento malý nedostatek tento software může autor dál vřele doporučit.

## 7 Seznam použité literatury

- [1] **Adaptic**, s.r.o. JavaScript. tvorba-webu.cz. [Online] Adaptic, s.r.o. [Citace: 1. 3. 2018.] <https://www.tvorba-webu.cz/javascript/>.
- [2] **Gault, Doug**. Beginning Oracle Application Express 5. New York : Apress, 2015. 978-1-4842-0462-2.
- [3] **Jacek Seń** . [Online] [Citace: 4. 10. 2016.] [http://www.jaceksen.pl/wordpress/wp-content/uploads/2011/03/110308\\_workspace.jpg](http://www.jaceksen.pl/wordpress/wp-content/uploads/2011/03/110308_workspace.jpg).
- [4] **Janovský, Dušan**. Kaskádové styly. Jak psát web. [Online] [Citace: 1. 3. 2018.] <https://www.jakpsatweb.cz/enc/kaskadove-styly.html>.
- [5] **Janovský, Dušan**. Úvod do JavaScriptu. Jak psát web. [Online] [Citace: 1. 3. 2018.] <https://www.jakpsatweb.cz/javascript/javascript-uvod.html>.
- [6] **Oracle**. Oracle Application Express Documentation. Oracle Help Center. [Online] <https://docs.oracle.com/database/apex-5.1/HTMDB/img/GUID-8FCA9B5B-7668-455A-812E-8BAB3D8B7C2B-default.png>.
- [7] **Oracle**. Oracle® Database - Quick Installation Guide. Oracle. [Online] [Citace: 25. 6. 2016.] [https://docs.oracle.com/cd/E11882\\_01/install.112/e49703.pdf](https://docs.oracle.com/cd/E11882_01/install.112/e49703.pdf).
- [8] **Oracle**. Application Express Installation Guide. Oracle Help Center. [Online] [Citace: 8. 6. 2016.] [https://docs.oracle.com/cd/E11882\\_01/install.112/e12196/pre\\_require.htm](https://docs.oracle.com/cd/E11882_01/install.112/e12196/pre_require.htm).
- [9] **Scott John, Buytaert Nick, Cannell Karen, D'Souza Martin a další**. Expert Oracle Application Express. místo neznámé : Apress, 2015. ISBN-13 (electronic): 978-1-4842-0484-9.

- [10] **Sciore, Edward.** Understanding Oracle APEX 5 Application Development, 2nd Edition. místo neznámé : Apress, 2015. ISBN-13 (electronic): 978-1-4842-0989-9.
- [11] **Shannon, Ross.** What is HTML? HTML Source. [Online] [Citace: 1. 3. 2018.] <http://www.yourhtmlsource.com/starthere/whatishtml.html>.
- [12] **Sharma, Manish.** Introduction to PL/SQL stored procedure in Oracle Database by Manish Sharma. Manish Sharma. [Online] [Citace: 2. 2. 2017.] <http://www.rebellionrider.com/pl-sql-tutorials/stored-procedure/introduction-to-pl-sql-stored-procedure>.
- [13] **Taskkill.** 1. díl - Úvod do AJAXu. ITnetwork.cz. [Online] [Citace: ] <https://www.itnetwork.cz/javascript/ajax/uvod-do-ajaxu>.
- [14] Co je vlastně Oracle APEX? Digitec Solutions. [Online] Digitec Solutions. [Citace: 11. 10. 2016.] <http://www.digitec.cz/cs/co-je-vlastne-oracle-apex>.
- [15] What is Oracle APEX? Oracle. [Online] Oracle. [Citace: 6. 4. 2017.] <http://www.oracle.com/technetwork/developer-tools/apex/overview/what-is-oracle-apex-3840637.html>.
- [16] CSS Introduction. w3schools.com. [Online] [Citace: 1. 3. 2018.] [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp).
- [17] SQL Tutorial. w3schools.com. [Online] [Citace: 1. 3. 2018.] <https://www.w3schools.com/sql/>.
- [18] What is PL/SQL? - Definition from Techopedia. Techopedia. [Online] [Citace: 1. 3 2018.] <https://www.techopedia.com/definition/24437/procedural-languagestructured-query-language-pl-sql>.

- [19] Application Express Installation Guide. Oracle Help center. [Online] Oracle. [Citace: 20. 3. 2016.]  
[https://docs.oracle.com/cd/E11882\\_01/install.112/e12196/toc.htm](https://docs.oracle.com/cd/E11882_01/install.112/e12196/toc.htm).
- [20] Get Oracle APEX access - Free WorkSpace, Download or Cloud. Techtown. [Online] [Citace: 5. 3. 2018.] <https://www.atechtown.com/install-oracle-application-express/>.
- [21] Oracle Application Express. Oracle. [Online] Oracle. [Citace: 2. 6. 2016.]  
<https://apex.oracle.com/en/>.
- [22] Chapter 6: Team Development - Agile Oracle Application Express [Book]. O'Reilly | Safari. [Online] [Citace: 4. 10. 2016.] view-source:<https://www.safaribooksonline.com/library/view/agile-oracle-application/9781430237594/Chapter06.html>.
- [23] Understanding Oracle Application Express Applications. Oracle® Application Express Application Builder User's Guide. [Online] Oracle. [Citace: 2. 2. 2017.]  
[https://docs.oracle.com/html/E39147\\_04/bldapp\\_wiz.htm](https://docs.oracle.com/html/E39147_04/bldapp_wiz.htm).
- [24] Establishing User Identity Through Authentication. Application Express Application Builder User's Guide. [Online] Oracle. [Citace: 3. 2. 2017.]  
[https://docs.oracle.com/database/121/HTMDB/sec\\_authentication.htm#HTMDB12003](https://docs.oracle.com/database/121/HTMDB/sec_authentication.htm#HTMDB12003).
- [25] Creating Tabs. Application Express Application Builder User's Guide. [Online] Oracle. [Citace: 2. 2. 2018.]  
[https://docs.oracle.com/database/121/HTMDB/nav\\_tabs.htm#HTMDB28458](https://docs.oracle.com/database/121/HTMDB/nav_tabs.htm#HTMDB28458).
- [26] DATE, TIMESTAMP and format masks. Application Express: tips, tricks and best practice. [Online] Oracle. [Citace: 3. 1. 2018.]  
<https://apex.oracle.com/pls/apex/germancommunities/apexcommunity/tipp/6381/index-en.html>.

- [27] Managing Pages in a Database Application. Application Express Application Builder User's Guide. [Online] Oracle. [Citace: 2. 2. 2018.]  
[https://docs.oracle.com/database/121/HTMDB/bldapp\\_pg.htm#HTMDB05003](https://docs.oracle.com/database/121/HTMDB/bldapp_pg.htm#HTMDB05003).
- [28] Runtime Developer Toolbar. Application Express App Builder User's Guide. [Online] Oracle. [Citace: 2. 3. 2017.]  
[https://docs.oracle.com/cd/E71588\\_01/HTMDB/runtime-developer-toolbar.htm#HTMDB29547](https://docs.oracle.com/cd/E71588_01/HTMDB/runtime-developer-toolbar.htm#HTMDB29547).
- [29] Modifying the Universal Theme by using Theme Roller in Application Express 5.1. Oracle Help Center. [Online] [Citace: 2. 3. 2017.]  
<http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/apex/r51/themeroller/themeroller.html>.
- [30] PL/SQL: Building with Blocks. Oracle. [Online] Oracle. [Citace: ]  
<http://www.oracle.com/technetwork/issue-archive/2011/11-mar/o21plsql-242570.html>.
- [31] Creating Forms. Application Express Application Builder User's Guide. [Online] Oracle. [Citace: 2. 2. 2018.]  
[https://docs.oracle.com/cd/E11882\\_01/appdev.112/e11947/bldapp\\_frm.htm#HTMDB05005](https://docs.oracle.com/cd/E11882_01/appdev.112/e11947/bldapp_frm.htm#HTMDB05005).
- [32] Controlling Access to Applications, Pages, and Page Components. Oracle® Application Express Application Builder User's Guide . [Online] Oracle. [Citace: 2018. 2. 4.] [https://docs.oracle.com/html/E39147\\_04/bldapp\\_acc.htm](https://docs.oracle.com/html/E39147_04/bldapp_acc.htm).
- [33] Using Page Designer. Oracle® Application Express Application Builder User's Guide. [Online] Oracle. [Citace: 2. 5. 2018.]  
[https://docs.oracle.com/html/E39147\\_04/pg\\_designer003.htm](https://docs.oracle.com/html/E39147_04/pg_designer003.htm).

## 8 Seznam obrázků

Obr. 1 vztah mezi uživateli, pracovními prostory a databázovými schématy. ....	11
Obr. 2 ukázka prostředí Instance Administration .....	12
Obr. 3 hlavní stránka s moduly .....	13
Obr. 4 ikona Application builder .....	14
Obr. 5 rozložení nástroje Page designer .....	15
Obr. 6 ikona SQL Workshop.....	16
Obr. 7 ikona Team Development.....	17
Obr. 8 ikona Packaged Applications.....	18
Obr. 9 možné typy nové aplikace.....	20
Obr. 10 ukázka pohledu na nově vytvořenou aplikaci.....	23
Obr. 11 UML class diagram znázorňující vztahy mezi aplikací, stranami a jejími typy ...	24
Obr. 12 ukázka toolbaru v testovacím módu aplikace .....	27
Obr. 13 ukázka funkce Quick edit .....	28
Obr. 14 ukázka pohledu na nově vytvořenou aplikaci.....	29
Obr. 15 ikona Form on a procedure.....	32
Obr. 16 ukázka tabulky autentizačních schémat.....	35
Obr. 17 ukázka stránky s řízením uživatelských přístupů .....	35
Obr. 18 ukázka modelu pro Access control .....	37
Obr. 19 ukázka struktury URL aplikace v APEX .....	38
Obr. 20 ukázka výpisu stavu session .....	39
Obr. 21 ukázka výukové aplikace .....	41

Univerzita Hradec Králové  
Fakulta informatiky a managementu  
Akademický rok: 2016/2017

Studijní program: Aplikovaná informatika  
Forma: Prezenční  
Obor/komb.: Aplikovaná informatika (ai3-p)

**Podklad pro zadání BAKALÁŘSKÉ práce studenta**

<b>PŘEDKLÁDÁ:</b>	<b>ADRESA</b>	<b>OSOBNÍ ČÍSLO</b>
Medek Petr	Bavlnářská 533, Semily - Podmoklice	I14104

**TÉMA ČESKY:**

Oracle Application Express

**TÉMA ANGLICKY:**

Oracle Application Express

**VEDOUCÍ PRÁCE:**

Ing. Barbora Tesařová, Ph.D. - KIKM

**ZÁSADY PRO VYPRACOVÁNÍ:**

cíl práce: Cílem práce je vytvoření výukových materiálů Oracle APEX pro podporu výuky.

Osnova práce:

1. Teoretická část
  - 1.1. Relační databáze
  - 1.2. Oracle
  - 1.3. Apex
2. Tvorba výukové aplikace
  - 2.1. Popis požadavků na systém
  - 2.2. Rozbor aplikačních modulů
3. Zhodnocení výsledků
4. Závěr

**SEZNAM DOPORUČENÉ LITERATURY:**

Dokumentace Oracle - <https://docs.oracle.com>

Sheldon, Robert. SQL začínáme programovat. Praha, 2005.

Pokorný, Jaroslav. Databázové systémy. Vyd. 2., přeprac. Praha, 2003. ISBN 80-01-02789-9

Sciore, Edward. 2015. Understanding Oracle APEX 5 Application Development, 2nd Edition. místo neznámé : Apress, 2015.

ISBN-13 (electronic): 978-1-4842-0989-9.

Podpis studenta: .....

Datum: .....

Podpis vedoucího práce: .....

Datum: .....