

Login xlapes02, **Lapeš Zdeněk**, 2. ročník BITP, prezenční, FIT
Ak. rok 2021/2022 studium č.2: řádný zápis

[Zpět na termíny](#)

[[IDS Home]]

Navigace

- [Hlavní stránka](#)
- [Poslední změny](#)
- [Nápověda](#)

Stránka

Nástroje

- [Odkazuje sem](#)
- [Nahrát soubor](#)
- [Seznam souborů](#)
- [Seznam stránek](#)
- [Historie stránky](#)

[Stránka](#)[Zdroj](#)[Historie](#)[Změny](#)[Sledovat](#)

Projekty v předmětu IDS

Obsah

- 1 [Popis projektu](#)
- 2 [1. část - Datový model \(ERD\) a model případů užití](#)
- 3 [2. část - SQL skript pro vytvoření objektů schématu databáze](#)
- 4 [3. část - SQL skript s dotazy SELECT](#)
- 5 [4. část - SQL skript pro vytvoření pokročilých objektů schématu databáze](#)
- 6 [5. část - Dokumentace popisující finální schéma databáze](#)
- 7 [Organizace projektu, řešení a obhajoby](#)
- 8 [Hodnocení řešení projektu](#)
- 9 [Školní databázový server Oracle](#)
- 10 [Doporučená literatura a pomůcky pro řešení projektu](#)
- 11 [Software](#)
 - 11.1 [Klienti při připojení k databázi Oracle](#)
 - 11.2 [Databázový stroj Oracle](#)
 - 11.3 [Vývoj klientských aplikací v prostředí Oracle](#)
 - 11.4 [Modelování](#)

Popis projektu

Cílem řešeného projektu je návrh a implementace relační databáze na zvolené téma. Je možné pokračovat na projektu z předmětu IUS. Pokud se studenti rozhodnou v projektu z IUS nepokračovat, pak si příslušné téma dvojice vybere ze [seznamu témat](#) .

Zvolené téma není potřeba předem nikde hlásit, avšak všechny výsledky musí obsahovat název zvoleného tématu (např. jako komentář v SQL skriptech, či vyznačený v diagramech nebo dokumentaci).

Projekt sestává ze pěti částí, které se odevzdávají ve stanovených termínech do WISu. Všechny odevzdávané SQL skripty musí být určené pro systém řízení báze dat (SŘBD) Oracle 12c a musí umožňovat opakované volání, tedy odstranění a opětovné vytvoření, nebo rovnou přepsání, objektů v databázi a jejich dat (je v pořádku, když při prvním volání skriptu proběhnou neúspěšné pokusy o odstranění neexistujících databázových objektů).

Část projektu s diagramy a část projektu s dokumentací se odevzdávají vždy jako jeden PDF soubor. Části projektu s SQL skripty se odevzdávají jednotlivě vždy v jednom souboru s prostým textem v kódování znaků UTF-8.

1. část - Datový model (ERD) a model případů užití

- Datový model (ER diagram) zachycující strukturu dat, resp. požadavky na data v databázi, vyjádřený jako diagram tříd v notaci UML nebo jako ER diagram např. v tzv. [Crow's Foot](#) notaci a model případů užití vyjádřený jako diagram případů užití v notaci UML reprezentující požadavky na poskytovanou funkcionalitu aplikace používající databázi navrženého datového modelu. Datový model musí obsahovat alespoň jeden vztah generalizace/specializace (tedy nějakou entitu/třidu a nějakou její specializovanou entitu/podtřidu spojené vztahem generalizace/specializace; vč. použití správné notace vztahu generalizace/specializace v diagramu).
- Odevzdává se dokument obsahující výše uvedené modely včetně stručného popisu datového modelu. Z popisu musí být zřejmý význam jednotlivých entitních a vztahových množin.

2. část - SQL skript pro vytvoření objektů schématu databáze

- SQL skript vytvářející základní objekty schématu databáze, jako jsou tabulky vč. definice integritních omezení (zejména primárních a cizích klíčů), a naplňující vytvořené tabulky ukázkovými daty. Vytvořené schéma databáze musí odpovídat datovému modelu z předchozí části projektu a musí splňovat požadavky uvedené v následujících bodech.
- V tabulkách databázového schématu musí být alespoň jeden sloupec se speciálním omezením hodnot, např. [rodné číslo](#) či [evidenční číslo pojištěnce \(RČ\)](#) , [identifikační číslo osoby/podnikatelského subjektu \(IČ\)](#) , [identifikační číslo lékařského pracoviště \(IČPE\)](#) , [ISBN](#) či [ISSN](#) , [číslo bankovního účtu](#) (vizte také [tajemství čísla účtu](#)), atp. Databáze musí v tomto sloupci povolit pouze platné hodnoty (implementujte pomocí CHECK integritního omezení).
- V tabulkách databázového schématu musí být vhodná realizace vztahu generalizace/specializace určená pro čistě relační databázi, tedy musí být vhodně převeden uvedený vztah a související entity datového modelu do schéma relační databáze. Zvolený způsob převodu generalizace/specializace do schéma relační databáze musí být popsán a zdůvodněn v dokumentaci.
- Skript také musí obsahovat automatické generování hodnot primárního klíče nějaké tabulky ze sekvence (např. pokud bude při vkládání záznamů do dané tabulky hodnota primárního klíče nedefinována, tj. NULL).

3. část - SQL skript s dotazy SELECT

- SQL skript, který nejprve vytvoří základní objekty schéma databáze a naplní tabulky ukázkovými daty (stejně jako skript v bodě 2) a poté provede několik dotazů SELECT.
- Konkrétně musí tento skript obsahovat alespoň dva dotazy využívající spojení dvou tabulek, jeden využívající spojení tří tabulek, dva dotazy s klauzulí GROUP BY a agregační funkcí, jeden dotaz obsahující predikát EXISTS a jeden dotaz s predikátem IN s vnořeným selectem (nikoliv IN s množinou konstantních dat). U každého z dotazů musí být (v komentáři SQL kódu) popsáno srozumitelně, jaká data hledá daný dotaz (jaká je jeho funkce v aplikaci).

4. část - SQL skript pro vytvoření pokročilých objektů schématu databáze

- SQL skript, který nejprve vytvoří základní objekty schéma databáze a naplní tabulky ukázkovými daty (stejně jako skript v bodě 2), a poté zdefiniuje či vytvoří pokročilá omezení či objekty databáze dle upřesňujících požadavků zadání. Dále skript bude

obsahovat ukázkové příkazy manipulace dat a dotazy demonstrující použití výše zmiňovaných omezení a objektů tohoto skriptu (např. pro demonstraci použití indexů zavolá nejprve skript EXPLAIN PLAN na dotaz bez indexu, poté vytvoří index, a nakonec zavolá EXPLAIN PLAN na dotaz s indexem; pro demonstraci databázového triggeru se provede manipulace s daty, která vyvolá daný trigger; atp.).

- Tento SQL skript musí konkrétně obsahovat vše z následujících
 - vytvoření alespoň dvou netriviálních databázových triggerů vč. jejich předvedení,
 - vytvoření alespoň dvou netriviálních uložených procedur vč. jejich předvedení, ve kterých se musí (dohromady) vyskytovat alespoň jednou kurzor, ošetření výjimek a použití proměnné s datovým typem odkazujícím se na řádek či typ sloupce tabulky (`table_name.column_name%TYPE` nebo `table_name%ROWTYPE`),
 - explicitní vytvoření alespoň jednoho indexu tak, aby pomohl optimalizovat zpracování dotazů, přičemž musí být uveden také příslušný dotaz, na který má index vliv, a v dokumentaci popsán způsob využití indexu v tomto dotazu (toto lze zkombinovat s EXPLAIN PLAN, vizte dále),
 - alespoň jedno použití EXPLAIN PLAN pro výpis plánu provedení databázového dotazu se spojením alespoň dvou tabulek, agregační funkcí a klauzulí GROUP BY, přičemž v dokumentaci musí být srozumitelně popsáno, jak proběhne dle toho výpisu plánu provedení dotazu, vč. objasnění použitých prostředků pro jeho urychlení (např. použití indexu, druhu spojení, atp.), a dále musí být navržen způsob, jak konkrétně by bylo možné dotaz dále urychlit (např. zavedením nového indexu), navržený způsob proveden (např. vytvořen index), zopakován EXPLAIN PLAN a jeho výsledek porovnán s výsledkem před provedením navrženého způsobu urychlení,
 - definici přístupových práv k databázovým objektům pro druhého člena týmu,
 - vytvořen alespoň jeden materializovaný pohled patřící druhému členu týmu a používající tabulky definované prvním členem týmu (nutno mít již definována přístupová práva), vč. SQL příkazů/dotazů ukazujících, jak materializovaný pohled funguje,
- Řešení projektu může volitelně obsahovat také další prvky neuvedené explicitně v předchozích bodech či větší počet nebo složitost prvků uvedených. Takové řešení pak může být považováno za nadstandardní řešení a oceněno dalšími body. Příkladem nadstandardního řešení může být řešení obsahující
 - klientskou aplikaci realizovanou v libovolném programovacím jazyce, přičemž práce s aplikací odpovídá případům užití uvedených v řešení 1. části projektu – tedy aplikace by neměla pouze zobrazovat obecným způsobem tabulky s daty a nabízet možnost vkládání nových či úpravy a mazání původních dat, ale měla by odpovídat pracovním postupům uživatelů (např. knihovník po příchodu čtenáře žádá ID průkazky čtenáře, systém vypíše existující výpůjčky čtenáře s vyznačením případných pokut, knihovník má možnost označit jednotlivé výpůjčky jako právě vrácené, případně inkasovat pokuty spojené s výpůjčkami, či přidat nové výpůjčky daného čtenáře),
 - SQL dotazy a příkazy ukazující transakční zpracování, vč. jejich popisu a vysvětlení v dokumentaci – např. ukázka atomicity transakcí při souběžném přístupu více uživatelů/spojení k jedněm datům, ukázka zamykání, atp.

Tip: pro ladění PL/SQL kódu v uložených procedurách či databázových triggerech můžete použít proceduru `DBMS_OUTPUT.put_line(...)` pro výstup na terminál klienta.

5. část - Dokumentace popisující finální schéma databáze

- Dokumentace popisující jednotlivé body řešení ze skriptu ve 4. části vč. jejich zdůvodnění (např. popisuje výstup příkazu EXPLAIN PLAN bez indexu, důvod

Organizace projektu, řešení a obhajoby

Studenti řeší projekt ve dvojici (v týmu). Každý z výsledků projektu musí být vypracován v souladu se studijními předpisy VUT a FIT a autorským zákonem, tj. zejména samostatně dvojicí studentů (týmem), která jej předkládá, jako svůj výsledek (viz čl. 11 Směrnice děkana FIT doplňující Studijní a zkušební řád VUT).

Pro řešení studenti využívají čas volného využití v počítačových učebnách CVT nebo řeší na svých počítačích. Cvičící poskytují zájemcům konzultace. Kromě toho jsou zařazena do programu přednášek témata na podporu řešení projektů zaměřená na seznámení s prostředím, které budou studenti využívat k řešení projektů, např. databázový server Oracle 12c, vývojové prostředí Oracle SQL Developer a jazyk PL/SQL – vizte přednášky.

Součástí 4. a 5. části je závěrečná obhajoba projektu. Cílem obhajob je zdůvodnit a diskutovat prezentované řešení a prokázat samostatnou práci (na obhajobě můžete být požádáni o vysvětlení či upřesnění kterékoliv části projektu).

Hodnocení řešení projektu

Za projekt lze získat celkem 34 bodů, z toho za poslední část lze získat až 19 bodů. Za výsledek obsahující další funkcionalitu či prvky nepožadované explicitně v zadání projektu lze získat další prémiové body.

Celkově lze dosáhnout nejvýše 34 bodů. Za jednotlivé části řešení je následující počet bodů:

1. *Datový model (ERD) a model případů užití* (s obhajobou) – max. 5 bodů
2. *SQL skript pro vytvoření základních objektů schématu databáze* – max. 5 bodů
3. *SQL skript s několika dotazy SELECT* – max. 5 bodů
4. *SQL skript pro vytvoření pokročilých objektů schématu databáze a Dokumentace popisující finální schéma databáze* (s obhajobou) – až 19 bodů

Školní databázový server Oracle

Přibližně v průběhu druhého týdne výuky budou všem studentů kurzu IDS vytvořeny uživatelské účty na školním databázovém server Oracle. Ve WISu bude termín "Přihlašovací údaje na školní databázový server Oracle" v jehož popisu budou informace o způsobu připojení na server a v komentáři k (nulovému) hodnocení heslo pro přihlášení (login je stejný, jako jinde na FIT).

O vytvoření účtů a možnostech přihlášení budete také informováni v hromadném emailu.

Doporučená literatura a pomůcky pro řešení projektu

- [Oracle Database 12c Release 1 \(12.1\) – Get Started](#)
- [Oracle Database Online Documentation 12c Release 1 \(12.1\) – Database SQL Language Reference](#)
- [Oracle Database Online Documentation 12c Release 1 \(12.1\) – Database PL/SQL Language Reference](#)
- [Oracle Database Online Documentation 12c Release 1 \(12.1\) – SQL Developer User's Guide](#)
- [Oracle Database Online Documentation 12c Release 1 \(12.1\) – SQL*Plus User's Guide and Reference](#)

- Ráb, J.: Tvorba databázových aplikací prostředí Oracle Form Builder. Studijní opora, 2006. Dostupné mezi soubory k předmětu ve složce Studijní opory.
- Visual Paradigm – Use Case Modeling
- Visual Paradigm – Entity Relationship Diagram
- Visual Paradigm – Class Diagram
- Jaroslav Ráb: Prezentace práce ve vývojovém prostředí Oracle Forms Builder 6i/10g.
 1. Uvod do tvorby databazovych aplikaci v prostredi Oracle Forms Builder 6i (wmv format/resolution 1024x768, filesize 240 MB, 45 minut)
 2. Prace s vice okny (wmv format/resolution 1024x768, filesize 90 MB, 15 minut)
 3. Prace s listboxem, dynamicke nastaveni DEFAULT_WHERE klauzule databazoveho data bloku (wmv format/resolution 1024x768, filesize 48 MB, 9 minut)
 4. Prace s tiskovou sestavou (wmv format/resolution 1024x768, file size 120 MB, 15 minut)

Software

Některý níže uvedený software je dostupný také přímo na tomto webu, bez nutnosti stahovat z webu Oracle.

Klienti při připojení k databázi Oracle

- Oracle SQL Developer
- Oracle SQL Developer Command Line SQLcl (SQL*Plus compatible)
- JetBrains DataGrip (free for students and teachers)

Databázový stroj Oracle

- Oracle Database 11g Express Edition
- Oracle Pre-Built Developer VMs (Oracle Enterprise Manager Developer VM)
- Oracle Docker Container Registry (Oracle Database SE2/EE, Instant Client)

Vývoj klientských aplikací v prostředí Oracle

- [Obraz virtuálního stroje pro Oracle VirtualBox s Windows XP a nástroji Oracle](#) (velikost 5 GB; vizte [návod](#)) – nástroje Oracle Forms a Reports Builder 10g Release 2, Oracle database 11gR2 XE, Oracle SQL Developer, Oracle Data Modeler, SQL Plus a další podpůrné nástroje
- [Obraz virtuálního stroje pro Oracle VirtualBox s CentOS 5.9 a nástroji Oracle](#) (velikost 4GB; vizte [návod](#))

Modelování

- Visual Paradigm for UML, Standard Edition – Academic Partner Program
- Visual Paradigm, Community Edition
- Obeo UML Designer (Eclipse)
- Modeliosoft Modelio
- PlantUML