

Projekty

Popis projektu

Cílem řešeného projektu je návrh a implementace relační databáze na zvolené [téma](#).

Projekt sestává ze pěti částí, které se odevzdávají ve stanovených termínech do WISu:

1. *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 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).
2. *SQL skript pro vytvoření základních objektů schématu databáze* – SQL skript vytvářející základní objekty schéma 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 [upřesňující požadavky zadání](#).
3. *SQL skript s několika 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 dle [upřesňujících požadavků zadání](#).
4. *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 zadefinuje 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.).
5. *Dokumentace popisující finální schéma databáze* – Dokumentace popisující řešení ze skriptu v bodě 4 vč. jejich zdůvodnění (např. popisuje výstup příkazu EXPLAIN PLAN bez indexu, důvod vytvoření zvoleného indexu, a výstup EXPLAIN PLAN s indexem, atd.).

Všechny odevzdávané SQL skripty musí být určeny 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.

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](#).

Po první části projektu následuje *obhajoba vytvořených modelů* a po poslední části *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).

Témata projektu

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 téma (např. jako komentář v SQL skriptech, či vyznačený v diagramech nebo dokumentaci).

Upřesňující požadavky zadání projektu

- 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 nebo TRIGGER).
- 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.
- SQL skript obsahující dotazy SELECT musí obsahovat konkrétně 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 konstatních dat). U každého z dotazů musí být (v komentáři SQL kódu) popsáno srozumitelně, jaká data hledá daný dotaz (jaká je jeho funkce v aplikaci).
- SQL skript v poslední části projektu musí obsahovat vše z následujících
 - vytvoření alespoň dvou netriviálních databázových triggerů vč. jejich předvedení, z toho právě jeden trigger pro automatické generování hodnot primárního klíče nějaké tabulky ze sekvence (např. pokud bude při vkládání záznamů do dané tabulky hodnota primárního klíče nedefinována, tj. NULL),
 - 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 navrhnout 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 prémiovými 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 jednomu 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.

Hodocení řešení projektu

Za řešení splňující všechny požadavky definované v [popisu částí projektu](#) a [upřesňujících požadavcích zadání projektu](#) lze získat celkem 29 bodů. Za nadprůměrný výsledek obsahující [další funkcionality či prvky nepožadované explicitně v zadání projektu](#) lze získat dalších 5 bodů, zde je ponechán prostor pro iniciativu dvojice.

Celkově lze dosáhnout nejvýše 29 až 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)* – max. 14 bodů, či až 19 bodů v případě naprůměrného výsledku.

Pro hodnocení poslední části projektu 15 až 19 body je nutné odevzdat nadstandardní řešení. Vizte [příklady výše](#) (na konci sekce [Upřesňující požadavky zadání projektu](#)).