

Zápočet:

1) Navrhnout (ERD) a vytvořit (SQL) DB, která se bude týkat libovolného tématu:

- a) musí obsahovat minimálně 7 tabulek s PK
 - nepřehánět s počtem tabulek (max. 15)
 - možné využít DB z předmětu KI/(K)URD, KI/(K)URDB
 - k zápočtu s sebou ERD databáze (pro snadnější orientaci)
- b) musí být naplněná daty
 - v průměru minimálně 20 záznamů na tabulku
 - např. mám 7 tabulek x 20 záznamů = alespoň 140 záznamů v celé db (ve všech tabulkách dohromady)
- c) musí být implementována (SQL)
 - student si zvolí vhodný DBMS (např. MySQL, MariaDB, PostgreSQL, MS SQL Server, Oracle, DB2, ...)

2) Připravit si příkazy nad vytvořenou DB z bodu 1):

- a) **SELECT (4x)**, varianty s poddotazem (select v selectu) a bez poddotazu
 - jeden SELECT vypočte průměrný počet záznamů na jednu tabulku v DB
 - jeden SELECT bude obsahovat vnořený SELECT
 - jeden SELECT bude obsahovat nějakou analytickou funkci (SUM, COUNT, AVG,...)
 - jeden SELECT bude řešit rekurzi nebo hierarchii (JOIN)
- b) **VIEW (1x)** s podstatnými informacemi z alespoň několika tabulek
 - alespoň tři tabulky, mezi informacemi nemají figurovat FK a nedůležité informace
 - pro spojení tabulek použijte různé typy příkazu **JOIN** (inner, left, right, natural, ...)
- c) **INDEX (1x)**, indexový soubor nad nějakým sloupcem tabulky
 - alespoň jeden netriviální indexový soubor (unikátní, fulltextový, ...)
- d) **FUNCTION (1x)**, která bude realizovat výpočet nějaké hodnoty z dat v DB
 - např. celkovou cenu objednávky, počet účastníků starších 65 let, průměrnou cenu nabízených produktů apod.
- e) **PROCEDURE (1x)**, která bude používat **1x CURSOR** a také **1x ošetření chyb** (HANDLER / TRY...CATCH / RAISE / EXCEPTION - dle zvoleného DBMS)
 - např. vytvoří a naplní novou tabulku informacemi o náhodných slevách na vybrané výrobky, nebo zákazníkům vygeneruje slevové bonusy podle určitých podmínek, apod.
- f) **TRIGGER (1x)**, který ošetří práci s daty nad nějakou tabulkou DB
 - využívají se hlavně pro příkazy INSERT, UPDATE, DELETE, ale můžete zkusit i jiný typ
 - např. pro UPDATE do nové tabulky zapíše datum, čas a informace o uživateli, který nějakým způsobem upravoval data v dané tabulce
- g) **TRANSACTION (1x)** použít v některé z předchozích procedur / funkcí
 - tj. uzavřít skupinu příkazů do transakce a ošetřit případ, kdy není možné všechny uvedené příkazy vykonat najednou
 - např. převod peněz z jednoho účtu na druhý uzavřít do transakce + ošetřit situaci kdy odesílatel nemá na účtu dostatek financí na provedení převodu
 - START TRANSACTION, COMMIT, ROLLBACK (případně i SAVEPOINT)
- h) **USER** - mít předem připravené příkazy na ukázkou
 - umět vytvořit/odstranit účet uživatele CREATE/DROP USER
 - umět se přihlásit jako právě vytvořený uživatel a ověřit dostupnost databázi
 - umět vytvořit/odstranit roli CREATE/DROP **ROLE** (některé DBMS nemají)
 - umět přidělit/odebrat uživateli nebo roli nějaká práva GRANT / REVOKE
- i) **LOCK** mít předem připravené příkazy na ukázkou
 - umět zamknout/odemknout tabulku (případně celou databázi, nebo jen řádek - pokud to zvolený DBMS umožňuje)
 - LOCK TABLE Vyroby READ / UNLOCK TABLE Vyroby / UNLOCK TABLES
- j) **ORM** – některý z výše uvedených úkolů realizovat pomocí vhodného ORM (SQLAlchemy)

POZOR: některé DBMS nepodporují používání procedur, v takovém případě využijte podporované programové bloky (funkce nebo metody).

Zápočtové práce budou individuálně kontrolovány (online nebo prezenčně) na zápočtových termínech ve zkouškovém období ZS (případně i dříve po dohodě s vyučující) - ideálně **do 8. 1. 2023**.
Termíny zápočtů budou uvedeny v systému STAG.

Aplikace pro navrhování databází:

Database Modeler Lite : <https://play.google.com/store/apps/details?id=adrian.adbm&hl=cs&gl=US>

Xampp - phpMyAdmin - Návrhář : <https://www.apachefriends.org/index.html>

MySQL Workbench : <https://www.mysql.com/products/workbench/>

DBeaver : <https://dbeaver.io/>

DataGrip : <https://www.jetbrains.com/datagrip/> (30 days trial version)

Diagram.DrawIO : <https://app.diagrams.net/> (online diagram sw)

a další.