

NoSQL – zadání projektu (**PROTOTYP**)

Projekty budou vypracovávat převážně dvojice studentů, v případě jednotlivce bude přihlédnuto k faktu, že byl na celý projekt sám.

Přihláška i samotný **výsledný projekt** se odevzdává formou „úkolu“ do **Olivy**.

Přihláška

Přihláška bude obsahovat:

1. Název práce
2. Jména studentů a jejich zodpovědnosti (kdo co bude dělat)
3. E-R diagram, ze kterého se bude vycházet (viz první bod požadavků níže)

Zadání

Forma odevzdané práce: ZIP archiv obsahující dokumentaci (HTML + obrázky), požadované skripty (prosté textové soubory) a jejich spool výstupy (prosté textové soubory).

Odevzdaný projekt se bude skládat z následujících částí:

1. Obrázek s modelem (E-R diagram) (**2B**)
 - Musí obsahovat alespoň 4 entity, lze použít model z vlastních předchozích semestrálních prací apod. Doporučena jsou originální zadání (nikoliv různé knihovny, databáze CD/DVD, autobazary apod.)
2. docker-compose s využitou (NoSQL) databází a vizualizačním nástrojem (**1B**)
3. Slovní popis validačních schémat definujících strukturu dokumentů a omezení hodnot jednotlivých atributů (/ klíčů) (**2B**)
 - Např. že plat musí být kladné číslo; že jméno musí být textová hodnota obsahující pouze písmena; že informace o (ne)splnění položky todo-listu musí být logická hodnota apod.
4. Návrh API rozhraní „*business logiky*“ pro alespoň dva procesy, jako např. přijetí zaměstnance = založení záznamu zaměstnance, svázání s nadřízeným, svázání se sdíleným služebním vozem (vazba N:M), založení požadavku na koupi pracovních pomůcek (jeden notebook, jedna myš). Procesy by měly být složitější než jen takové, které vedou na jeden insert či update dokumentu. (**2B**)

- Např. vytvoření záznamu, uložení vráceného id do proměnné, využití id v proměnné při vytvoření/editaci dalšího/jiného záznamu apod.
5. Návrh (alespoň) pěti slovně formulovaných dotazů nad schématem – musí se jednat o různorodé netriviální dotazy (navrhněte dotazy vedoucí na použití spojení kolekcí, množinových operací, agregací atd.) **(3B)**
 6. Skript, který vytvoří databázové schéma odpovídající E-R diagramu **(3B)**
 - Skript bude obsahovat validační schémata definující strukturu dokumentů a omezení hodnot jednotlivých atributů dokumentu
 7. Skript, který naplní kolekce (testovacími) daty **(1B)**
 - Kolekce by měly obsahovat řádově tisíce dokumentů
 - Pro generování testovacích dat můžete použít různé již existující nástroje
 8. Skript (včetně výstupu spool), který provede postupně všechny navržené dotazy (viz výše) **(5 x 2B)**
 - Z výstupu bude patrný dotaz, jeho exekuční plán (`explain()`) a výsledek
 - U dvou (dle úvahy) nejsložitějších dotazů se pokuste vymyslet ještě druhou verzi dotazu (vracející tentýž výsledek), a porovnáním exekučních plánů (např. podle `executionTimeMillis` nebo `totalDocsExamined`), které budou součástí výstupu posuďte, který dotaz byl efektivnější
 - Pro výše zvolené dva dotazy vytvořte (složený) index pro odpovídající klíč(e) (dle úvahy) a posuďte, který z dotazů je díky jakému indexu efektivnější
 - Pro (alespoň) dva dotazy (dle úvahy) využijte agregační pipeline
 - Každá pipeline musí obsahovat alespoň dvě etapy
 9. Skript, který se pokusí porušit postupně veškeré nastavené validace atributů dokumentů + výstup spool z provádění tohoto skriptu **(2B)**
 - Ve výpise budou vidět chyby informující o porušení validace, zaměřte se na chybovou hlášku – popis validace příslušného atributu
 10. ??? TODO - **Potenciální replikace**??? **(2B)**
 - Pozn. Prozatím zahrnuto do bodového hodnocení
 11. Skript, který zálohuje celou databázi **(1B)**
 - Včetně výstupu spool

12. Skript pro vyčištění (/ vymazání) všech databází (**1B**)