**🔎**  **Úkol č. 13 - Databáze (12b)**  
  
Naše databáze se skládá ze 3 tabulek:  
Jedná se o fiktivní databázi obchodu  
ER diagram - pod textem.

***První tabulka je produc*t:**ta drží aktuální informace o produktech.  
**Sloupce**  
**product\_id:**Text, Unique, Non null, primary key -- naše interní ID produktu  
**title:**Text, non null -- název produktu  
**price:**Float, non null -- cena za kterou produkt prodáme nyní, vždy bez 21 % DPH. Do sazby 21% spadají všechny naše produkty  
**description:**Text -- popis produktu  
**brand:**Text -- výrobce produktu  
**stock\_quantity:** Integer -- počet produktů skladem

***Druhá tabulka:***

**product\_marketing\_data** (*Dávej si pozor tahle tabulka nemusí obsahovat všechny produkty z tabulky product, některé produkty nám právě přišly na sklad a ještě na ně neběží reklamní kampaň.)*

**row\_id:**Integer, PK -- jen primární klíč pro potřeby optimalizace tabulky  
**product\_id:**String, FK -- id produktu ke kterému se data vztahují  
**clicks:** Integer -- kolikrát klienti na produkt v reklamě kliknuli  
**impressions:**Integer -- počet zobrazení reklam s produktem  
**spend:** Float -- útrata na reklamách za produkt

***Třetí tabulka:***

**product\_sales\_data** (D*ávej si pozor tahle tabulka nemusí obsahovat všechny produkty z tabulky product, některé produkty nikdo nekupuje.)*  
**row\_id:**Integer, PK -- jen primární klíč pro potřeby optimalizace tabulky  
**product\_id:** String, FK -- id produktu ke kterému se data vztahují  
**item\_quantity:** Integer -- kolikrát se produkt prodal  
**item\_revenue:**Float -- za kolik peněz se všechny produkty prodaly, cena se mohla měnit v čase, proto to nemusí odpovídat sale\_price \* item\_quantity  
  
  
Soubor s předchystanou databází si můžeš stáhnout zde <https://finalexam.trnavsky.eu/final_exam.db>.  
A pak jej nahrát do oblíbeného prostředí třeba <https://sqliteonline.com/> které jsme používali na lekci.  
V případě použití SQL syntaxe jiné, než SQLite to prosím někde u odpovědi zaznač.  
  
**Tvým úkolem bude sestrojit tyto queries:**  
*1) Zjisti počet položek, součet všech kusů a průměrnou cenu produktu, co jsou skladem, výsledné sloupce ve výsledku pojmenuj "pocet\_polozek" resp. "pocet\_kusu", "prumerna\_cena\_skladu" (3b.)*

počet položek: SELECT COUNT(product\_id) AS pocet\_polozek FROM product ;

součet všech kusů: SELECT SUM(stock\_quantity) AS pocet\_kusu from product ;

průměrná cena produktu: SELECT AVG(price) AS prumerna\_cena\_skladu from product ;

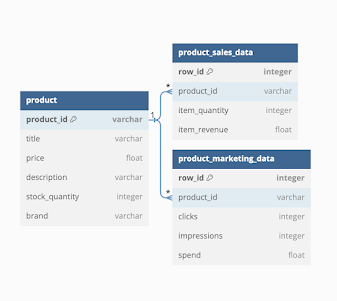
*2) Vypiš retailer\_id, title, aktuální cenu včetně DPH, pro všechny produkty které stojí 50 až 500 bez DPH včetně těchto hraničních hodnot (3b.)*

Sloupec retailer\_id není definován. Předpokládam že se myslí product\_id. SELECT product\_id, title, price \* 1.21 AS price\_with\_vat FROM product WHERE price >= 50 AND price <= 500;

*3) Vypiš title, brand, price, clicks a impressions****všech****produktů. Výsledek seřaď podle počtu kliknutí(clicks), kde největší počet kliknutí bude první (6b.)*

Použiji LEFT JOIN přotože chci zobrazit všechny produkty. Ty produkty které nejsou v tabulce product\_marketing\_data budou mit hodnotu clicks a impressions NULL. Pokud bychom chtěli zobrazit pouze produkty které mají nějaký click a impressions použila bych INNER JOIN nebo RIGHT JOIN (každá položka z tabulky product\_marketing\_data musí mít i záznam v tabulce product).

SELECT product.title, product.brand, product.price, product\_marketing\_data.clicks, product\_marketing\_data.impressions from product LEFT JOIN product\_marketing\_data on product.product\_id = product\_marketing\_data.product\_id ORDER BY product\_marketing\_data.clicks DESC;



ER Diagram