# Zadání RDB 2024.

1. Vytvořte databázi, která bude ukládat a zpracovávat data z mýtných bran.
2. Systém eviduje pro každé vozidlo seznam průjezdů v podobně JSON souboru následujícího formátu:

{

brana\_id : "1234",

typ\_brany: "mobilni, staticka, virtualni",

prujezd : {

datum\_prujezdu: "datetime",

registrace\_vozidla: {

"vozidlo": { "spz","typ vozidla", "hmotnost", "emisní třída","km\_sazba"},

"firma": {"nazev","ico","adresa"}

},

"ujete\_km" : "number"

}

systemova\_data: {

brana\_id : "1234",

verze\_os : "1.23",

systemovy\_datum\_cas: "timestamp",

{virtualni -> GPS: "GPS", presnost:"+/-metry"

staticka -> napeti: "V", proud: "A"

mobilni -> misto: "adresa", pozice: "GPS", baterie\_typ: "dle typu se stanovuje max pocet cyklu a jmenovite napeti", pocet\_cyklu: "10"}

}

1. Pro každou registraci vozidla ukládáme:
   1. kredit (Kč), historie plateb (typ platby (karta (číslo karty, planost, vlastník) | převod (číslo účtu, kód banky) | hotovost), datum platby, kredit)).
2. Připravte:
   1. Odpovídající databázové schéma pro relační databázi
   2. Jednoduchý způsob vložení nové registrace a plateb (UI, konzole, import dat)
   3. V relační databázi uložte strukturovaná data
   4. V nosql uložte zdrojová data o průjezdech a systémových dat
   5. Funkcionalitu:
      1. Výpočet zbývajícího kreditu na základě ujetých km (vypočtených dle ujetých vzdáleností mezi bránami).
      2. Aktualizace záznamu vozidla, pokud přijde jiná hodnota v JSON
      3. Vytvořit report o průjezdu vozidla v rámci stanovených dnů (celkový počet km, celková cena + plný formátovaný (ne JSON) export průjezdů bran z nosql databáze.
3. Pro relační databázi, je třeba použít <https://liquibase.org/>, pro případné změny ve schématu databáze.
4. ~~Řešte změny schématu i pro nerelační db.~~
5. Data budou generována pomocí offline generátoru (napsaný v C).