

GymBeam

Case Study - Data Engineer



Zadanie je navrhnuté tak, aby preverilo vaše schopnosti v nasledujúcich oblastiach:

- Dátové modelovanie
- Integrácia údajov
- Optimalizácia dátových kanálov
- Čistenie dát
- Práca s Pythonom a SQL
- Znalosti osvedčených postupov v oblasti dát

Dôležité informácie

- Pre riešenie niektorých úloh si budete musieť vytvoriť **bezplatný účet** na platforme **Keboola** (keboola.com).
- **Po ukončení všetkých úloh** nezabudnite pridať do svojho projektu nasledujúcich hodnotiteľov:
 - jakub.uhrina@gymbeam.com
 - ladislav.safarik@gymbeam.com

Držíme vám palce!

Ukážte svoje zručnosti a kreativitu pri riešení úloh. 😊

1. Návrh dátového modelu pre e-commerce platformu

Predstavte si, že pracujete pre e-commerce spoločnosť, ktorá predáva produkty online. Vašou úlohou je navrhnuť optimálny dátový model, ktorý bude podporovať:

1. **Správu produktov a ich kategórií.**
2. **Zákaznícke objednávky.**
3. **Históriu transakcií.**
4. **Analýzu predajov podľa rôznych dimenzií (napr. čas, produkt, kategória, región).**

Zadanie:

1. **Navrhnite dátový model vo forme ER diagramu (Entity-Relationship diagram), ktorý pokryje nasledovné požiadavky:**
 - a. **Produkty:** ID, názov, cena, opis, dostupnosť, kategória.
 - b. **Kategórie:** ID kategórie, názov kategórie, nadradená kategória (hierarchia kategórií).
 - c. **Zákazníci:** ID zákazníka, meno, email, adresa (vrátane regiónu), dátum registrácie.
 - d. **Objednávky:** ID objednávky, zákazník, dátum objednávky, stav objednávky, položky objednávky.
 - e. **Položky objednávky:** ID položky, produkt, množstvo, cena za jednotku.
 - f. **Transakcie:** ID transakcie, objednávka, dátum, spôsob platby, suma.
2. **Definujte dimenzie a faktové tabuľky pre analytické potreby:**
 - a. Navrhnite dátový model (napr. star schema alebo snowflake schema), ktorý bude slúžiť na analýzu predajov podľa času, produktov, kategórií a regiónov.
3. **Identifikujte:**
 - a. Primárne a cudzie klúče.
 - b. Možné normalizačné kroky a úrovne normalizácie.
 - c. Miesta, kde by denormalizácia mohla zvýšiť výkonnosť analytických dotazov.
4. **Pripravte SQL schému na vytvorenie tabuľiek podľa vášho návrhu.**

Dodatočné otázky na diskusiu:

- Aké kompromisy by ste spravili medzi normalizáciou a výkonnosťou?
- Ako by ste riešili historické zmeny (napr. zmena ceny produktu, adresa zákazníka)?
- Aké indexy by ste pridali na zlepšenie výkonnosti dotazov?

2. Integrácia dát

Vašou úlohou je navrhnuť a implementovať generický extraktor na spracovanie dát z API Golemio o mestských knižničiach.

Požiadavky:

- **Vstupy:** Otvorené dáta z platformy **Golemio** o mestských knižničiach.
- **Výstupy:** Extrahované dáta by mali obsahovať nasledujúcich 10 parametrov:
 1. **ID knižnice**
 2. **Názov knižnice**
 3. **Ulica**
 4. **PSC**
 5. **Mesto**
 6. **Kraj**
 7. **Krajina**
 8. **Zemepisná šírka**
 9. **Zemepisná dĺžka**
 10. **Čas otvorenia**

Odporúčaná možnosť: Použiť **Keboolu** na implementáciu generického extraktora a spravovanie aktualizácií dát prípadne **AWS** pomocou služieb ako: **AWS Lambda,Amazon S3,Amazon EventBridge,Amazon CloudWatch**

Alternatíva: Implementovať vlastný skript (napr. v **Python**) na stiahnutie a spracovanie dát. Následne nasadiť tento skript do verejného repozitára na **GitHub**, aby bolo riešenie dostupné na kontrolu.

Ďalšie požiadavky:

- Naplánujte extraktor tak, aby sa dáta aktualizovali **denne o 7:00 ráno (pražského času)**.
- Ak použijete vlastný skript, pridajte README súbor s inštrukciami na spustenie a konfiguráciu.

Dokumentácia API:

Kompletnú dokumentáciu API nájdete na tomto odkaze:

[https://api.golemio.cz/docs/openapi/#/%F0%9F%8F%A2%EF%B8%8F%20Municipal%20Libraries%20\(v2\)](https://api.golemio.cz/docs/openapi/#/%F0%9F%8F%A2%EF%B8%8F%20Municipal%20Libraries%20(v2))

3. Manuálny input

V projekte **Keboola** vytvorte nový bucket v úložisku (Storage) a pomenujte ho **manual-input**. V tomto buckete pridajte tabuľku s názvom **csv_input**.

Na jej vytvorenie použite súbor **input.csv**, ktorý vám bol doručený emailom.

Následne vytvorte **SQL transformáciu**, kde pomocou dotazov vyriešite problémy týkajúce sa kvality dát v tabuľke **csv_input**.

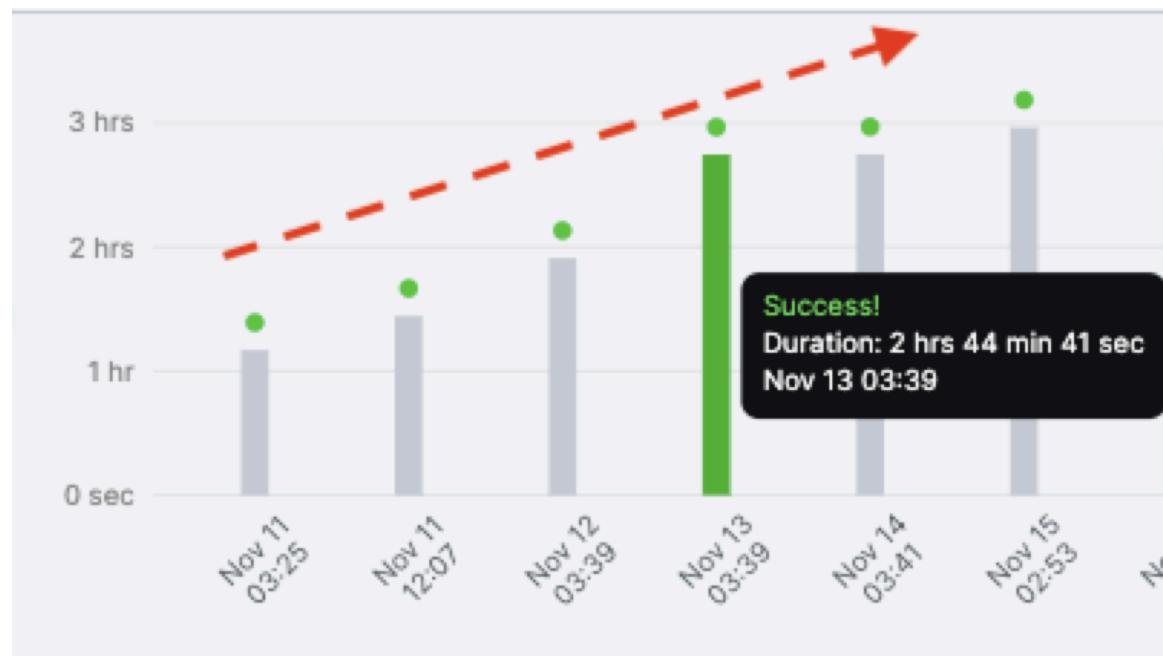
4. Výkonnostný problém v SQL transformácii

Kolega nasadil SQL transformáciu do produkcie. Spočiatku fungovala dobre, no časom sa doba spracovania výrazne predĺžila.

Úloha:

Na základe vašich skúseností identifikujte najčastejšie príčiny tohto správania a navrhnite možné riešenia.

Budťte konkrétni a načrtnite praktické kroky na odstránenie problému.



5. Otázky na hodnotenie osvedčených postupov

Zistite, či sa nižšie uvedené tvrdenia považujú za osvedčený postup. A ak nie, aký negatívny výsledok by mohol nastať a čo by ste odporučili ako lepšie riešenie? Pridajte prosím svoje osobné skúsenosti.

1. Používanie hardcoded hodnôt v ETL procesoch pre biznis pravidlá.
2. Neindexovanie stípcov, ktoré sú často dotazované vo veľkých tabuľkách.
3. Ukladanie logov a záloh na rovnaký server ako produkčná databáza.
4. Používanie zdieľaných servisných účtov na pripojenie k databázam v ETL nástrojoch.
5. Budovanie dátových kanálov bez implementácie mechanizmov na opakovanie alebo zotavenie pri zlyhaní.
6. Povoľovanie priameho prístupu ku zdrojovým dátam všetkým členom tímu bez kontroly prístupu.
7. Vynechanie validácie schémy pri načítavaní externých dát.
8. Používanie zastaraných ETL procesov bez pravidelných revízií optimalizácie.
9. Nepremazanie alebo neodstránenie zastaraných tabuľiek a pohľadov z dátového skladu.
10. Nenastavenie upozornení na zlyhané úlohy alebo oneskorenia kanálov.
11. Ukladanie citlivých údajov bez šifrovania pri ukladaní alebo prenose.
12. Ignorovanie obmedzení dátových typov pri vytváraní schém v dátovom sklade.
13. Povoľovanie kruhových závislostí medzi ETL úlohami.
14. Vykonávanie transformácií priamo na produkčných databázach namiesto staging vrstiev.
15. Výber dátového modelu (napr. hviezdica vs. snehová vločka) bez zohľadnenia použitia.
16. Používanie VARCHAR(MAX) ako predvoleného dátového typu pre textové polia.
17. Pridávanie všetkých stípcov zo zdrojového systému do dátového skladu bez ohľadu na ich relevantnosť.
18. Vynechanie partitioningu alebo clusteringu pre veľké faktové tabuľky.
19. Vývoj a nasadenie zmien v pipeline bez verzovania alebo testovania.