USE CASE · HŰSÉGPROGRAM · BESZÉLGETÉSALAPÚ ADATELEMZÉS

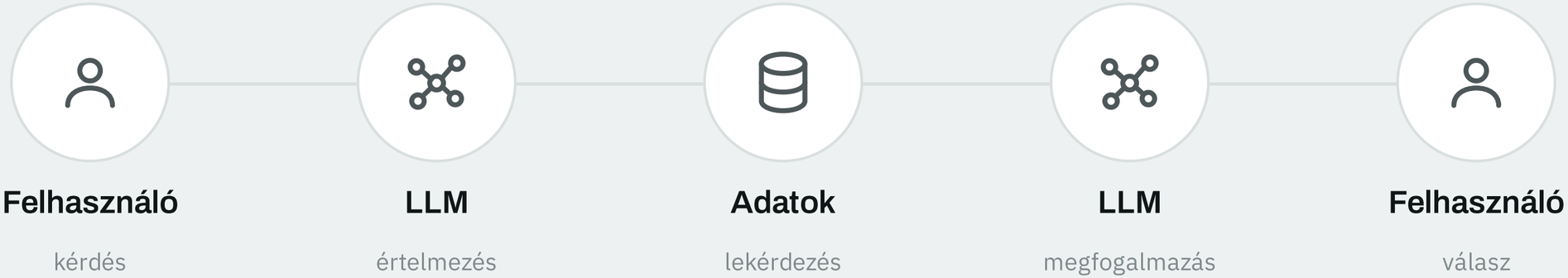
Conversational analytics.

Beszélgess az (SQL) adatbázisoddal.

UDVARI ISTVÁN DATA ANALYST

Dataklub Datashots prezentáció · 2026. 10. 28.

Egy darab adattáblát az ember is, az AI is átlát.



Egy adattábla: a modell a fejlécből is kitalálja, mit kell számolni, esetleg egyszerűen lehet irányítani.

Egy sok adattáblás adatbázissal már nem ilyen egyszerű a helyzet.

Az esettanulmányunkban szereplő adatbázis jellemzői

Ez egy hűségprogram adatbázisa. A hűségprogram alap mechanizmusa: a programtagok pontokat gyűjtenek, és a gyűjtött pontokért külső partnerek kedvezménykuponjait tudják megvásárolni.

~300
adattábla és view
operatív (OLTP) adatbázis, rengeteg bekódolt adattal és kódolótáblákkal

~2 000
adatmező

~270 M
adatsor
a legnagyobb adattábla ~80 M

75 GB
az adatbázis mérete
heti mentéseket kapunk

Nem elemzésre épült és nincs dokumentációja

Az objektumok nagy része az ügyfélportál működéséhez, az adatátadáshoz és a kampánymotorhoz kell. A jelentés a fejekben és régi lekérdezésekben él, nem az adatbázis sémában.

Az adattábla és adatmező nevekből nem kitalálható

Ha a modell megkapja a teljes adatbázis sémát, a nevekből, esetleg az adatokból találgat. Van olyan adattábla, ami nem feltétlenül az, aminek a neve alapján látszik.

Adatok és AI: ma nem téma. Az alapszabály: a személyes adatot eltávolítjuk, anonimizáljuk vagy átkódoljuk. A modellhez csak az adatbázis séma leírása és aggregált eredmény megy.

A modell nem csak az adatbázist nem érti, bennünket sem egyértelműen.

Réteg	Ahogy megjelenik
Ahogy az adattábla tárolja	`eseménytípus = 6`
View-ban feldolgozott érték	„Ajánlat vásárlás"
Ahogy beszélünk róla	**„letöltés"**

2,0×

eltérés a „hány aktív tagunk van?" kérdés két, egyaránt jogos olvasata között

~196 000 vs. ~100 000

Hány belépőnk volt a múlt hónapban?

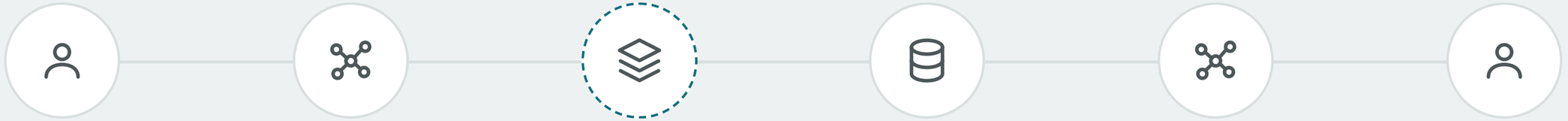
Regisztrációra vagy loginra gondolunk?

Értelmező réteg nélkül ez a kétértelműség észrevétlenül a válaszba kerül, magabiztos mondatban, jó formában.

És nem derül ki, amíg valaki utána nem számol.

Ha két eltérő időpontban kérdezzük ugyanazt, eltérő választ kaphatunk (sztochasztikus kimenet).

Komplex adatbázis: a kérdés és az adat közé kell egy értelmező lépés — a szemantikai réteg.



Mi a szerepe és miért hasznos? Adattárházi környezetből származó leírás:

Absztrakció
Elrejti a végfelhasználók és az elemzők elől a mögöttes adatbázis összetett szerkezetét (pl. kulcsokat, bonyolult SQL-joinokat).

„Egyértelmű igazságforrás” (Single Source of Truth)
Biztosítja, hogy egy-egy mérőszámot (pl. „belépések száma” vagy „aktív ügyfelek száma”) mindenki ugyanúgy értelmezzen a szervezetben.

Üzleti nyelvezet
Az olyan technikai oszlopnevek helyett, mint `usr_t_04_id`, érthető neveket használ (pl. „Vásárló azonosítója”).

Eszközfüggetlenség
Lehetővé teszi, hogy különböző BI- és elemző eszközök (például a Power BI szemantikai modelljei) ugyanabból az egységes definíciós bázisból dolgozzanak.

A szemantikai réteg: dokumentáció, amit ember és gép is olvas



Hogyan építhető fel a szemantikai réteg? Adatkörönként, beszélgetéssel.

FŐBB ADATKÖRÖK

- Ügyfél-törzs: státusz, szegmens, hozzájárulás
- Ajánlat-események: megtekintés, letöltés
- Ajánlat-törzs: kibocsátó, kategória, pontérték
- Promóciós kódok
- Kommunikáció: email, push
- Belépés
- Pontgyűjtés
- Adatfrissítés, ETL

A betanítás forrásai, megbízhatóság szerint besorolva

Forrás	Tag	Mire erős
Korábbi SQL-kódok (~590 fájl), view-definíciók	`kód`	Mit csinálunk; a „miért" gyenge.
Fejlesztői vagy megbízói írásos megerősítés	`ügyfél`	Mindent felülír.
Lefuttatott feltáró lekérdezés eredménye	`futás`	Tény, de időpontfüggő.
Az elemző szóbeli és írásos megerősítése	`István`	A saját folyamatainkra.
Üzleti leírások, egyeztető emailek, munkajegyzetek	`jegyzet`	Ellenőrizendő, lehet elavult.

A menet adatkörönként

Kódból indulunk → beszélgetés, feltárás → a modell összeírja, mi nem egyértelmű → ember dönt: megerősít vagy kizár → mérés → rögzítés forrással.

Egyszerűnek indult — nem volt az.

6–8

váratlan kérdés adatkörönként

Nem felesleges kérdések voltak. Megmutatták, mennyi kontextus van a fejünkben, amit a munka közben észre sem veszünk, hogy használunk.

Angolul: **tacit knowledge**. Ennek a kinyerése a munka nagyobbik fele.

Hol tárolódik a tudás? Szöveges fájlokban.

100%
md fájlokban
nincs finomhangolás, nincs vektoradatbázis

~35
md fájl a teljes körben
tudás, munkaszervezés, kimeneti formátumok, riportok

300 → 80
objektum maradt
16 átnézési blokk, egy blokk egy ülés

- Olvasható és ellenőrizhető: minden állítás mellett forrás és validációs státusz.
- Modellfüggetlen: bármelyik LLM megkaphatja, a tudás nem a modellben van.

Ami csendben romlik el, sokkal drágább, mint ami hangosan

elavult

Elavult adat és kód

Mezők, amik még ki vannak töltve, de évek óta nem frissülnek. Régi lekérdezések, amik olyan feliratra szűrnek, amit a rendszer már nem ad ki: egy oszlop csendben mindig 0.

concept drift

A jelentés elcsúszik

Egy negyedévig szinte nem ment kampánylevél, a push vitte a kommunikációt. Egy belépési mód-váltás óta a regisztráció dátuma mást jelent. Ugyanaz a kérdés előtte és utána mást mér.

időbeliség

Mikori adatbázis van nálunk?

A leíró adatok időbelisége

- Jelen- vagy múltbeli az információ?
- Időpontra vagy időszakra vonatkozik?

csendes nulla

A hiba, ami nem jelzett

Egy ékezetes szöveg rossz kódolással ért a szerverre: a szűrés félrement, hibaüzenet nélkül. A célcsoport lekérdezés hibás lett volna.

A válasz: hangos hiba > csendes tévedés. Minden tanulságból kód lett, nem jegyzet: kódolás-őr, „van esemény, nincs találat" őr, frissesség-őr. Ha egy szám elavult mezőre épülne, a rendszer nem ad számot.

Guardrails: az önállóságot szándékosan szűkre szabtuk

Ha az állítás forrása…	Dzsarvisz…
megerősített (fejlesztő, elemző)	magabiztosan állítja
lefuttatott lekérdezés, élő kód	állítja, a forrással
régi jegyzet	csak jelzéssel
két forrás ütközik	mindkettőt felajánlja
elavult	soha nem használja
nincs a rétegben	**„nem tudom"** antihallucináció

Az LLM önhatalmúlag, önállóan nem próbál az adatbázisból infókat „kiokoskodni".

Zárt világ

Csak dokumentált adattábla és adatmező. Az adatbázis séma katalógusa nem forrás: a nevekből következtetni tilos.

Csak olvasás, ember dönt

Kódszintű őr, csak SELECT. Definícióról a modell nem dönt: felveti a kétértelműséget.

48 / 48

tesztkérdés zárult a szándék szerint, ebből **13 helyes elutasítás**

Egy kérdés útja, lépésről lépésre

Szám, két olvasattal | Helyes elutasítás | Hangos hiba | ↻

DZSARVISZ · BESZÉLGETÉS

Hány aktív programtagunk van?

~196 000

programtag a legutóbbi heti adatállapot szerint.

Ha „aktív" alatt a lazább aktivitás-jelzőt érted, az ~100 000 főt ad — a kettő kb. kétszeres különbség, ezért nem választok helyetted.

`adatbázis frissessége: heti mentés` `forrás: futás` `definíció: megerősítve`

02 **Értelmezés**

mutató: **programtag** · időszak: jelen állapot · szűrő: technikai felhasználó nélkül

`„aktív" kétértelmű`

03 **Szemantikai réteg**

„aktív programtag" = státusz szerinti programtag `megerősítve`

másik olvasat: a laza aktivitás-jelző, kb. 2× különbséggel — ezt nem választja magától

04 **Lekérdezés + őr**

```
SELECT COUNT(*) AS programtag
FROM   ugyfel_nezet
WHERE  statusz_kod = 1
  AND  technikai   = 0;
```

`✓ csak SELECT` `✓ dokumentált objektum` `✓ kódolás rendben`

05 **Futtatás**

leválasztott futtató · 0,4 s · aggregált eredmény: 1 sor, azonosító nélkül

06 **Válasz**

szám + az adatbázis frissessége + olvasat + a másik olvasat

Leválasztott futtatás: a modell soha nem látja az élő adatbázist

modell oldal

Tesztadatbázison épül

A szemantikai réteg alapján SQL-szöveget ír, indoklással és forrásokkal. Az élő adatbázishoz nincs hozzáférése.

a határ

Csak két dolog lépi át

A modelltől: az SQL-szöveg, amit ember vagy kódőr ellenőriz, és futtatás előtt megmutat.
A modellhez: az aggregált eredmény, azonosító nélkül.

éles oldal

Független futtató

Csak olvasási joggal, a modelltől függetlenül futtatja a lekérdezést, és naplózza.

Minden válasz magával hozza: **az adatbázis frissességét**, **a használt definíciót** (és a másik olvasatot, ha van), **a lefutott lekérdezést** és **a korlátot**.

Hetekből napok — és csetelés az adatbázissal

FELADAT	HAGYOMÁNYOSAN		SZEMANTIKAI RÉTEGGEL · BECSÜLT
Új riport összeállítása	több nap – 1-2 hét	→	**~1 nap**
Egyszerű klaszter-modell egy adatkörből	több hét	→	**~1 nap**
Komplex elemzés pl. churn-előrejelzés, ügyfélérték-index	legalább 1 hónap	→	**~3 nap**

Példák a komplex körből: ügyfélérték-index · ajánlat-scorecard · email-fatigue · viselkedési klaszterek · push-hatás természetes kontrollcsoporttal · soft churn · next-best-offer · kampánytölcsér

> **Az admin felületen túl:** a kollégák az adatbázis szerkezetének ismerete nélkül kérdezhetnek, és a kimenet rögtön a célrendszer formátumában jön (CSV vagy Excel, kódolás, elválasztó).

Becslés a saját munkaidő alapján · tipikus értékek, nem kontrollált mérés

Amit ez a projekt megtanított

- **A nehézség nem az SQL kód megírása, hanem a jelentés.** „Aktív", „letöltés": mást jelent az adatban, a riportban és a folyosón.
- **A szemantikai réteg megállapodás, tudás, nem kód.** Olvasható fájlok, forrással; bármelyik modell megkaphatja.
- **Az AI-t keretek közé kell szorítani** a megfelelő viselkedés érdekében.
- **Az AI kérdései fárasztóak, de értékesek.** Felszínre hozzák, amit észrevétlenül tudunk.
- **A jó rendszer meg meri mondani, mit nem tud.** A „nem tudom" indokolt válasz.
- **Kell egy gazda.** A definíciók karbantartása is nagyon fontos. A változás állandó, az üzleti környezet, az adatbázis is folyamatos változásban van.

UDVARI ISTVÁN DATA ANALYST

Köszönöm a figyelmet!

Udvari István · Data Analyst

LINKEDIN linkedin.com/in/istvanudvari