



INFINITE
LAMBDA

Migráció felhőbe Snowflake alapokon

Your specialist **data engineering** partner

www.infinitelambda.com



Zoltan Csonka
data architect



Passionate and experienced data architect

I have many years experience with modelling, designing, building, migrating data warehouse and BI solutions. I have worked for multinational enterprises and small companies too in the field of insurance, banking, e-commerce, marketing, education.

Certified Snowflake professional

I am currently at **Infinite Lambda**, we are building cloud based data warehouse solutions to help our clients to make transform to data driven organisation.



We're hiring!

A feladat háttéréről

Az ügyfélről

Sikeres mobil alkalmazás fejlesztő angol cég.

A jelenlegi projekt egy gyerekeknek szóló, hang alapú, mobilos alkalmazásuk adatait dolgozza fel.

Az alkalmazás célja, hogy segítsen a gyerekeknek feszültséget oldani, segítse az elalvást.

- Világsztárok mondanak történeteket, meséket
- Altató, meditációs zenék, nyugtató hangok

A projekt célja

Segíteni az ügyfelet, hogy hatékonyabb, adatvezérelt vállalattá váljon.

Cél, hogy automatizált, modern adat platformot, riporting és elemző rendszert építsünk.

A mobil alkalmazással kapcsolatos

- használati,
- marketing,
- értékesítés segítő

riportokat és elemzéseket támogassuk.

A kihívás, problémák

Ami megvolt

- excel alapú kalkulációk, riportok
- külső alkalmazások adatait eltérő formában és úton importálták excelbe
- kézi folyamat

Ami nem jó

- ott kell lenni a fejlesztőnek
- nehezen átlátható, módosítható
- kézi feldolgozás miatt sok a hiba



Ami nem volt

- módszertani alapjuk
- excelen kívül más technológia
- nem volt történetiség
- nyers adat

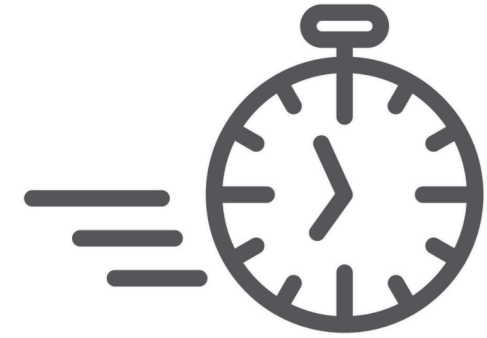
Követelmények - 1. fázis

- adjon end-to-end **keretet a továbbfejlesztéshez**
- napi jelentés **automatizálása**,
- a 31 KPI emigrálása (főként előfizetés és kampány adatok),
- BI eszközben elérhető napi jelentés **riport**

- **Könnyen módosítható**, gazdaságos legyen
- Oktatás, hogy a fejlesztést később belső kollégákkal folytathassák

Tulajdonosi elvárás a közel valós idejű riportok kiszolgálására alkalmas rendszer!

A megvalósítás határideje



4 hét

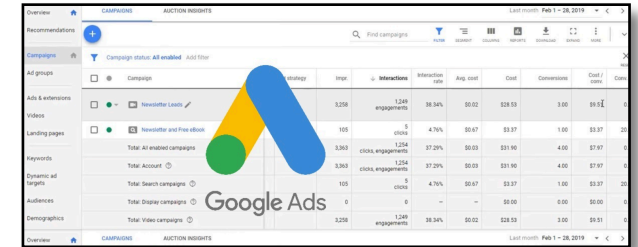
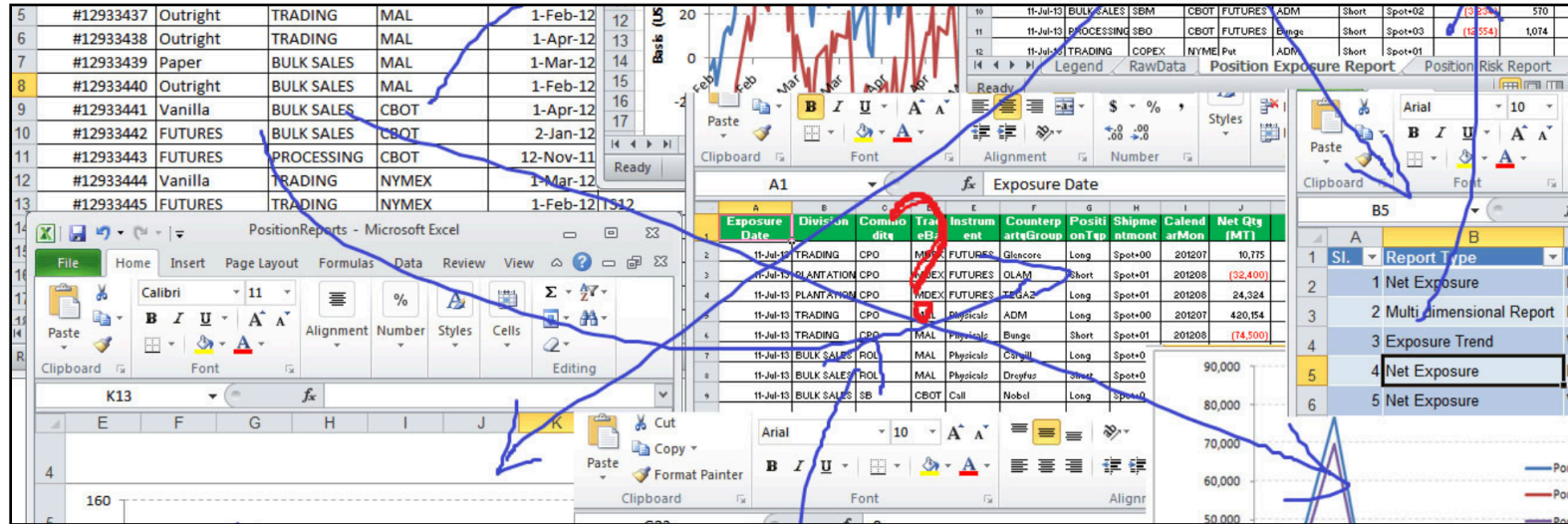
(+1 felmérés, előkészítés)

A projekt módszertana

Agilis, scrum, ticketing, minden hét végére leszállítandó és demo

Hol kezdjük?

A meglévő folyamatok és excel kalkulációk feltérképezése



(Subs)!\$M:\$M,'App Figures (Subs)!\$K:
 Adjust Installs & Trials!\$S:\$S,'Adjust Installs
 & Trials!\$J:\$J,"<="&G\$8,'Adjust Installs & Trials!\$J:
 \$J,">"&G\$8-7)+SUMIFS('Google & BT Acqs!\$M:\$M,'Google &
 BT Acqs!\$J:\$J,G\$8)

Az architektúra és a folyamat

Architecture - Data Lake

DATA LAKE

- Historikusan tárolt nyers adatok
- Tisztított, kiterített adatok
- Üzleti logika nélkül

**NEM VESZÍTHETÜNK ADATOT,
MINDEN ADATOT LÁTNI AKARUNK!**

The screenshot displays a data lake interface. On the left, a schema browser shows a tree structure under 'RAW_DEV' with various folders like 'ADJUST', 'APPFIGURES', 'APPLE_SEARCH_ADS_2_FT', 'AWS_MARKETPLACE', 'BASE', 'BRAINTREE_FT', 'FACEBOOK_AD_ACCOUNTS', 'FACEBOOK_AD_INSIGHTS', 'FACEBOOK_INSIGHTS_2', 'FACEBOOK_INSIGHTS_3', 'FACEBOOK_INSIGHTS_F', 'GOOGLE_ADS_FT', and 'UAC_EVENT_REPORTS'. A 'Details' window is open, showing a JSON payload for a query result. Below the details, a 'Data Preview' table is visible with columns 'Row', 'PAYLOAD', and 'FILE_KEY'. The table contains three rows of data.

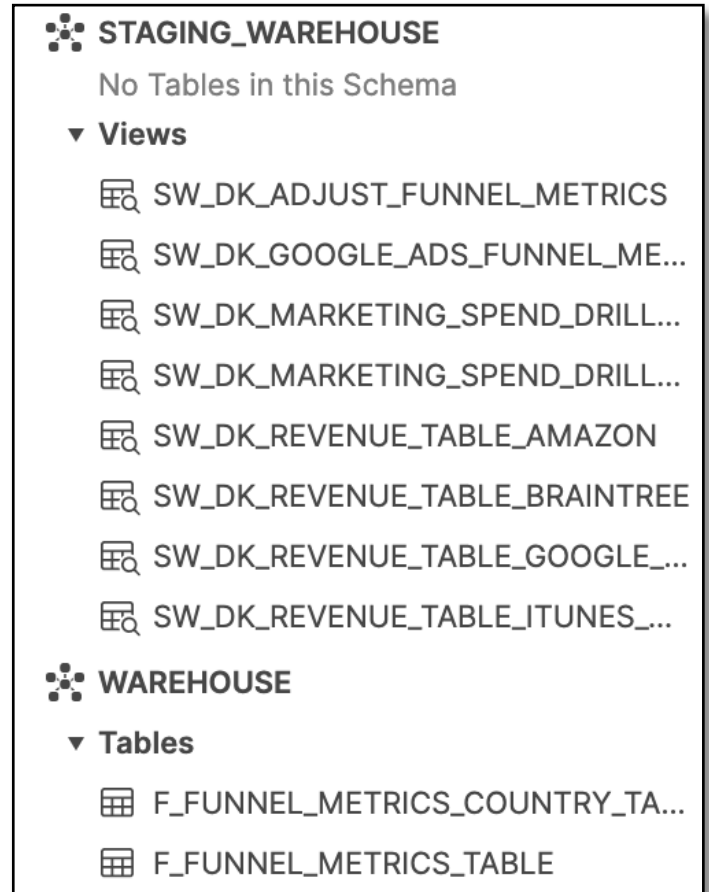
```
1 {
2   "params": {
3     "end_date": "2020-01-07",
4     "format": "json",
5     "group_by": "product, country",
6     "start_date": "2020-01-07"
7   },
8   "query_result": {
9     "280512153221": {
10      "AE": {
11        "average": "4.58",
12        "breakdown": [
13          1
```

Row	PAYLOAD	FILE_KEY
1	{ "params": { "end_date": "2020-01-07", "format": "json", ...	Appfigures/RATINGS/dt=2020-01-07/f245f71a-
2	{ "params": { "end_date": "2020-03-27", "format": "json", ...	Appfigures/RATINGS/dt=2020-03-27/e39f6de7
3	{ "params": { "end_date": "2020-03-31", "format": "json", ...	Appfigures/RATINGS/dt=2020-03-31/e36fac8b

Architecture - Warehouse

Data Warehouse

- Historikusan feldogozott adatok
- Üzleti KPI-ok
- Dimenzionális modell BI-hoz



STAGING_WAREHOUSE
No Tables in this Schema

▼ **Views**

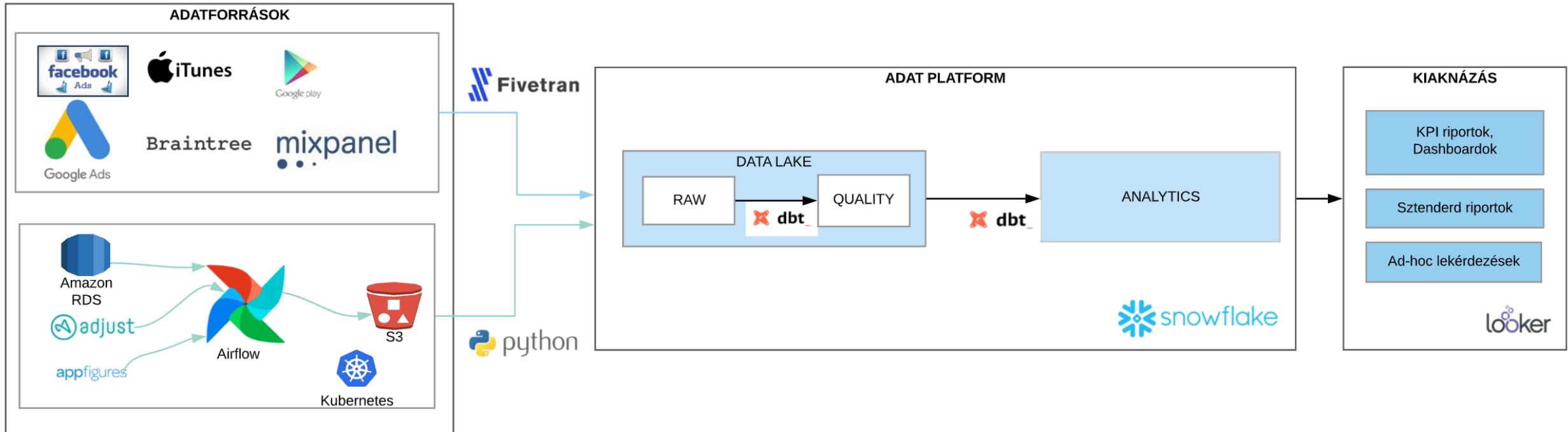
- SW_DK_ADJUST_FUNNEL_METRICS
- SW_DK_GOOGLE_ADS_FUNNEL_ME...
- SW_DK_MARKETING_SPEND_DRILL...
- SW_DK_MARKETING_SPEND_DRILL...
- SW_DK_REVENUE_TABLE_AMAZON
- SW_DK_REVENUE_TABLE_BRAINTREE
- SW_DK_REVENUE_TABLE_GOOGLE...
- SW_DK_REVENUE_TABLE_ITUNES...

WAREHOUSE

▼ **Tables**

- F_FUNNEL_METRICS_COUNTRY_TA...
- F_FUNNEL_METRICS_TABLE

Egyszerűsített architektúra



Fivetran **python**
Adatkinyerés, betöltés

dbt
Transzformáció

snowflake
Adatbázis, platform

Apache Airflow
Folyamatvezérlés

Az eszközök

Data warehouse, data lake platform



Felhőre fejlesztett relációs adatbázis

SQL alapú adatbázismotor

Egy platformon data lake, data warehouse

Számítások és adattárolás szétválasztott

Fantasztikus dokumentáció, rengeteg esettanulmány, pezsgő közösség

The screenshot displays the Snowflake web interface. The top navigation bar includes 'Databases', 'Shares', 'Data Marketplace', 'Warehouses', 'Worksheets', and 'History'. The main area shows a SQL query being executed in a worksheet. The query is a SELECT statement with various columns and a lateral flatten function. Below the query, the 'Results' section shows a 'Data Preview' table with 3 rows and 4 columns: 'Row', 'DATE', 'FILE_KEY', and 'PRODUCT_ID'. The results table is as follows:

Row	DATE	FILE_KEY	PRODUCT_ID
1	2020-01-07	Appfigures/...	280512153221
2	2020-01-07	Appfigures/...	280512153221
3	2020-01-07	Appfigures/...	280512153221

Below the results, there is a 'Warehouses' section with a table listing various warehouses. The table has columns for 'Status', 'Warehouse Name', 'Size', and 'Runn...'. The warehouses listed are:

Status	Warehouse Name	Size	Runn...
Suspended	WAREHOUSE_REPORT_DEV	X-Small	0
Started	WAREHOUSE_REPORT_PROD	X-Small	1
Suspended	WAREHOUSE_TRANSFORM_DEV	X-Small	0
Suspended	WAREHOUSE_TRANSFORM_PROD	X-Small	0
Suspended	WAREHOUSE_INGEST_DEV	X-Small	0
Suspended	WAREHOUSE_INGEST_PROD	X-Small	0

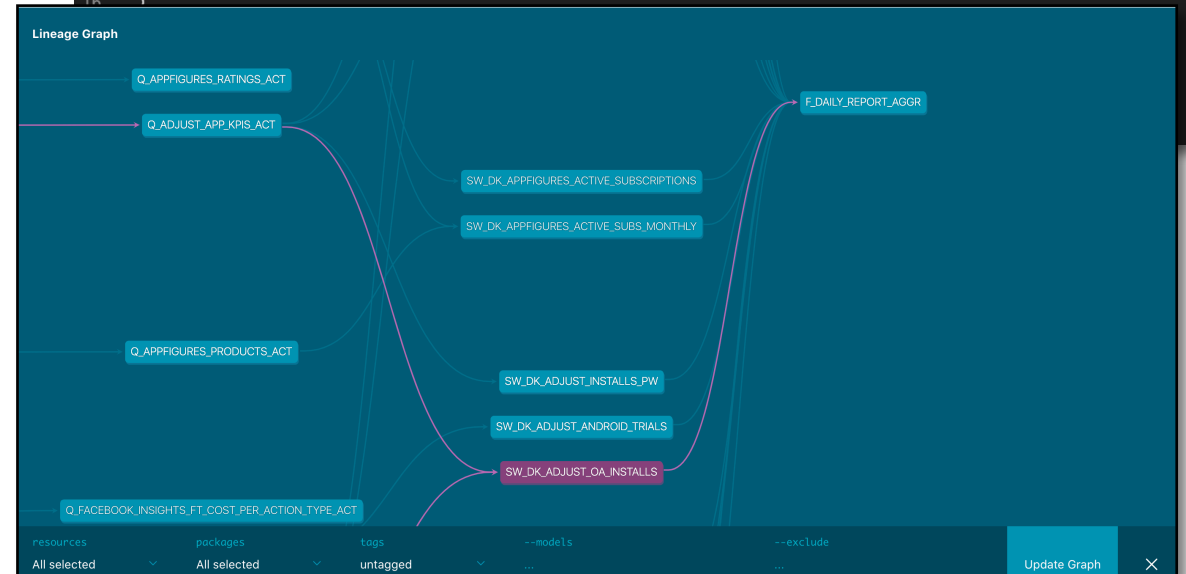
Transformation - ELT



dbt az adat transzformációk eszköze

- CTAS (SQL)
- Lineage (függőség kezelés)
- Dokumentációs segítség
- Open source

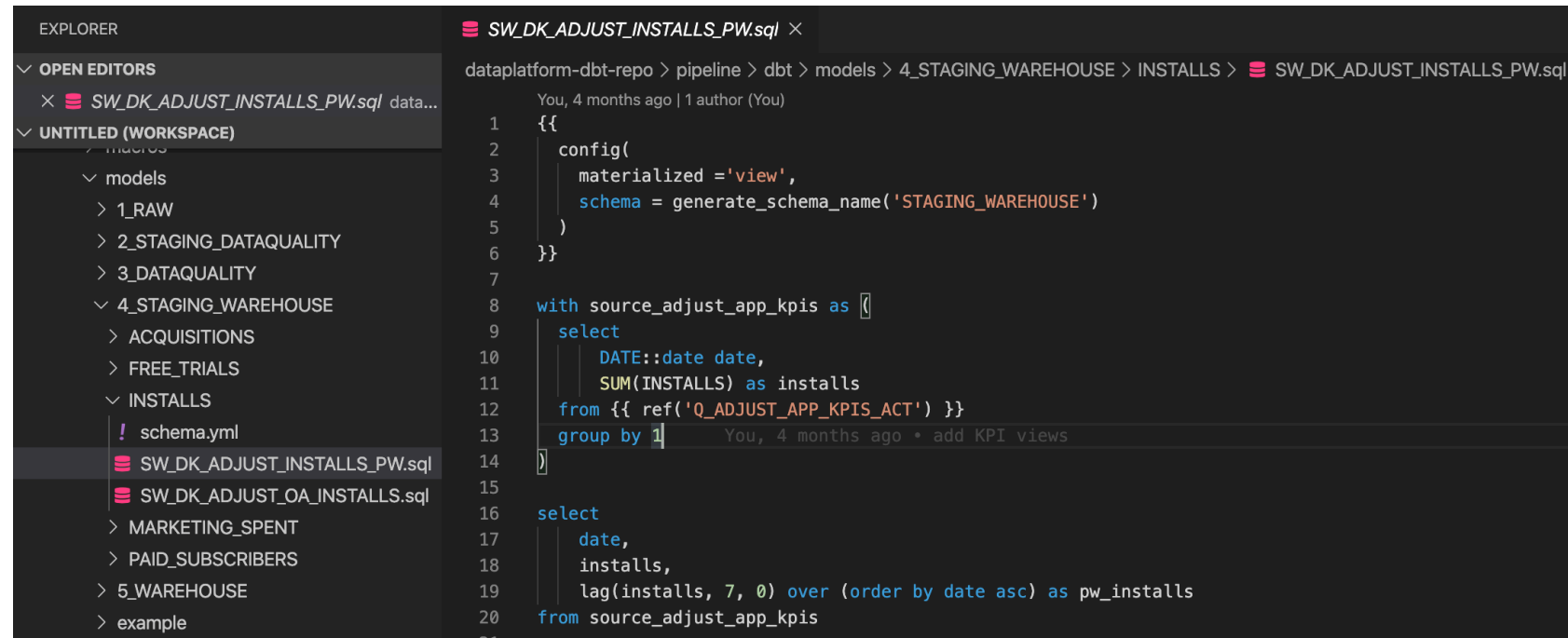
```
1  {{ You, 4 months ago * add KPI views
2  config(
3    materialized = 'view',
4    schema = generate_schema_name('STAGING_WAREHOUSE')
5  )
6  }}
7
8  with source_adjust_events as (
9    select
10     events_kpis.date,
11     SUM(case when network_groups.GROUP_TYPE = 'Group1' then events_kpis.SUBSCRIPTION_PAID_INIT_YHVRJP_E
12     SUM(case when network_groups.GROUP_TYPE = 'Group2' then events_kpis.SUBSCRIPTION_PAID_INIT_YHVRJP_E
13   from {{ ref('Q_ADJUST_EVENTS_KPIS_ACT') }} events_kpis
14   left join {{ ref('Q_ADJUST_NETWORK_GROUPS') }} network_groups on events_kpis.NETWORK = network_groups.N
15   group by 1
16 )
```



Transzformációk

DBT struktúra

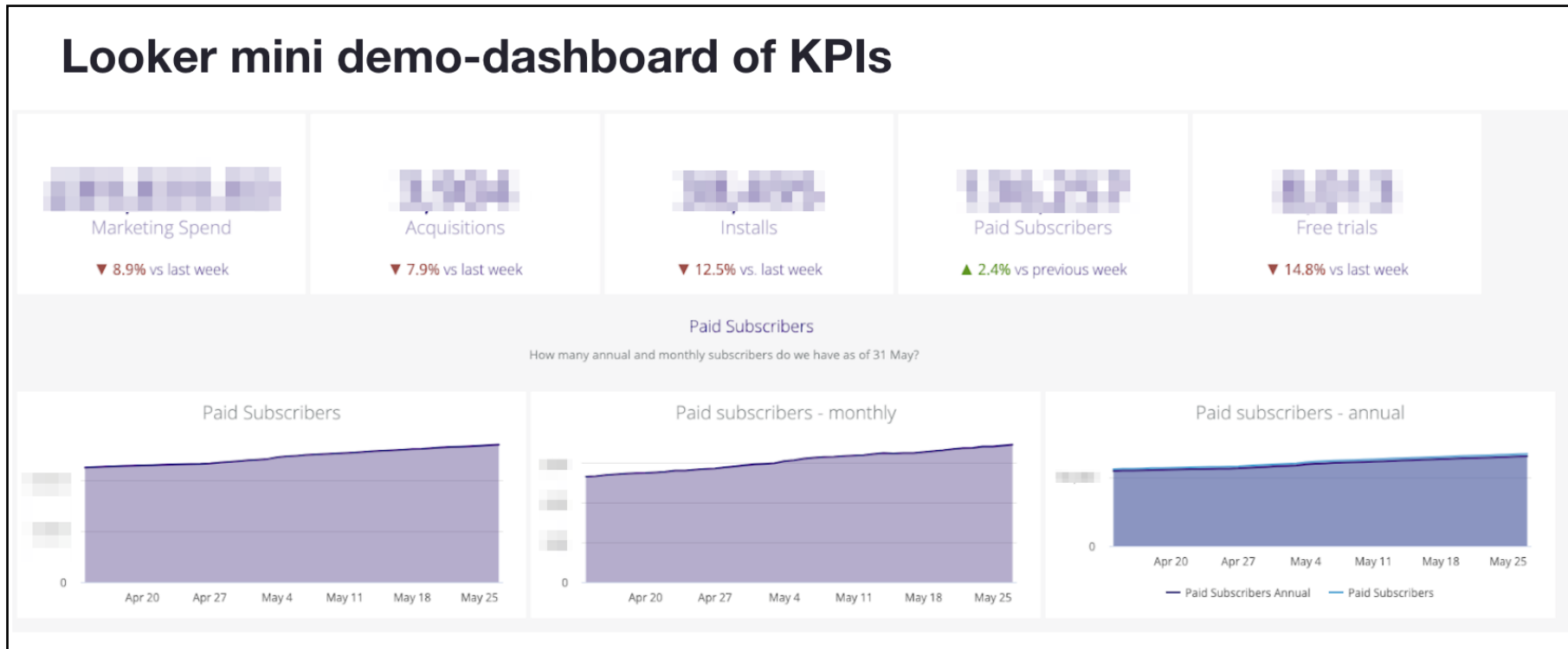
- Funkcionális rétegek
- KPI definíciók



The screenshot shows a DBT project explorer on the left and a SQL editor on the right. The explorer shows a directory structure with folders for 'models' and 'INSTALLS', and a file 'SW_DK_ADJUST_INSTALLS_PW.sql' selected. The editor shows the following SQL code:

```
dataplatfrom-dbt-repo > pipeline > dbt > models > 4_STAGING_WAREHOUSE > INSTALLS > SW_DK_ADJUST_INSTALLS_PW.sql
You, 4 months ago | 1 author (You)
1  {{
2  config(
3    materialized = 'view',
4    schema = generate_schema_name('STAGING_WAREHOUSE')
5  )
6  }}
7
8  with source_adjust_app_kpis as (
9    select
10     DATE::date date,
11     SUM(INSTALLS) as installs
12   from {{ ref('Q_ADJUST_APP_KPIS_ACT') }}
13   group by 1
14 )
15
16 select
17   date,
18   installs,
19   lag(installs, 7, 0) over (order by date asc) as pw_installs
20 from source_adjust_app_kpis
```

KPI riportok



Néhány kiváló Snowflake funkció, amit a projektben használtunk

Snowflake feature - semi-structured adat kezelése

VARIANT általános adattípus

- Variant type 16 MB bármilyen adattípust tartalmazhat (pl. JSON, AVRO, ORC, Parquet, XML forrásokból)
- DATA LAKE építéshez
- Séma változások kezelése végtelenül egyszerű
- Data science feladatokhoz
- SQL-ben egyszerűen lekérdezhető

The screenshot shows the Snowflake Data Preview interface. A 'Details' window is open, displaying a JSON payload:

```
1 {
2   "params": {
3     "end_date": "2020-01-07",
4     "format": "json",
5     "group_by": "product,country",
6     "start_date": "2020-01-07"
7   },
8   "query_result": {
9     "280512153221": {
10      "AE": {
11        "average": "4.58",
12        "breakdown": [
13          1

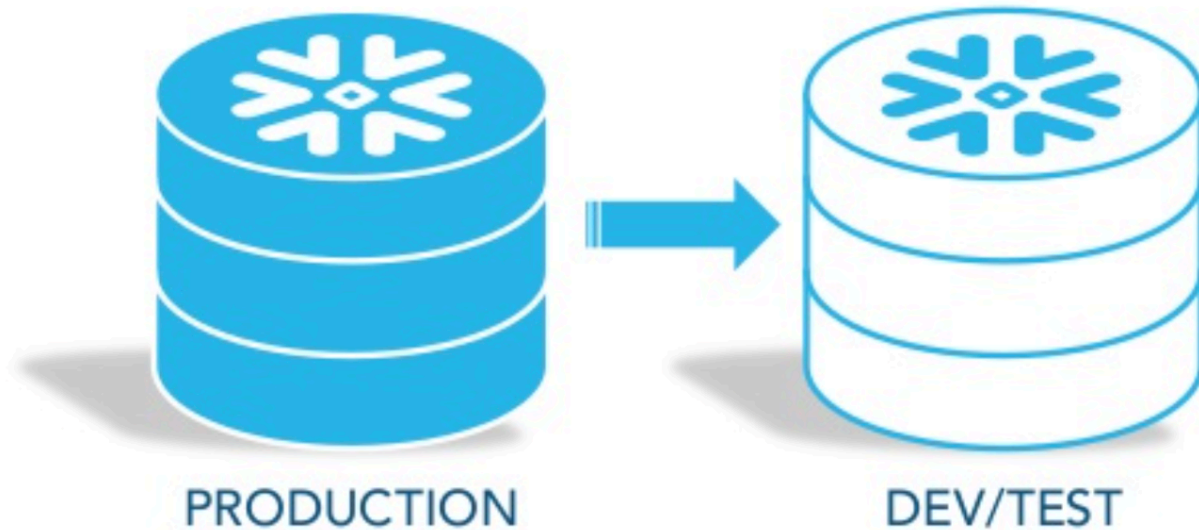
```

Below the details window, a table view shows the following data:

Row	PAYLOAD	FILE_KEY
1	{ "params": { "end_date": "2020-01-07", "format": "json", ...	Appfigures/RATINGS/dt=2020-01-07/f245f71a-
2	{ "params": { "end_date": "2020-03-27", "format": "json", ...	Appfigures/RATINGS/dt=2020-03-27/e39f6de7
3	{ "params": { "end_date": "2020-03-31", "format": "json", ...	Appfigures/RATINGS/dt=2020-03-31/e36fac8b

Snowflake feature - Zero copy clone

Teszt környezet létrehozása ingyen és azonnal



- táblákat, sémákat vagy egész adatbázist klónozik azonnal
- Nincs tárterület igény
- Az éles környezetet nem terheli
- A jogosultság örökölt
- Bármennyi készíthető

Snowflake feature - tuning és optimalizálás

- Többnyire nincs rá szükség
*nem kell és nem is lehet indexelni vagy hintelni, (és ezért én nagyon hálás vagyok)
partícionálni is csak ritkán szükséges (1TB táblaméret fölött)*
- Separate WH, cache, auto scale out, scale up

Lehet rossz SQL-t írni, hibás folyamatokat lefejleszteni itt is :)

Snowflake feature - Időutazás

Törölt és módosított adatok lekérdezése, visszaállítása

- táblákat, sémákat vagy egész adatbázist visszaállíthatunk
- 1-90 nap között
- van tárolási költsége



```
undrop table subscriptions;
```

```
select * from campaigns  
at(timestamp => 'Mon, 01 May 2020 16:20:00 -0700'::timestamp);
```

Összefoglalás

Nehézségek a projekt során

- Google, apple, amazon (is) eltérő definíciókat használ, eltérően számolja a mutatókat. pl. a letöltés darabszámot
- Fivetran replikáció nem tudja milyen adatot kap, a sorrendiség kezelése nem egyértelmű *pl. apple tranzakciók nem megfelelő sorrendben érkeznek*
- Némelyik forrásrendszer csak aggregált adatokat ad
- Looker riportokat nehéz volt Excellé varázsolni (a Looker sem Excel)

Leghasznosabb előnyök a projektből az ügyfél számára

- Teljesen automatikus folyamatok
- End-to-end keret, rugalmas módszertan és eszközök a továbbfejlesztéshez
- Nem kell adminisztrálni az eszközöket!
- Népszerű és könnyen használható eszközök. Snowflake, DBT, Airflow
- Könnyű a közös munkavégzés, megoszthatóak a folyamatok, adatok.

Nincs varázslat, működik!



Felhő migrációs gondolatok

- Felhő vagy sem, az a fontos, hogy ne az eszközzel kelljen küzdeni
- Nem kérdés, hogy Felhőben dolgozunk
- Hogyan kerül az adat a Felhőbe? *(ahol nincs delta az fáj itt is)*
- Felhő migrációnál a legnagyobb félelem még mindig az adatbiztonság
- Felhőben használjunk felhőre tervezett eszközöket

Még mindig jó megérteni mi kell az üzletnek, mit csinálunk.

**Olyan rendszert építünk,
ami rugalmas és
könnyen alkalmazkodik a változáshoz,
mert holnap már más lesz az igény!**



INFINITE
LAMBDA

Köszönöm a figyelmet!

Van kérdésed?

Contact us

info@infinitelambda.com

www.infinitelambda.com



INFINITE
LAMBDA

Agile, Cloud Based Data Solutions

We're hiring!

