**ETL & Data Pipeline**

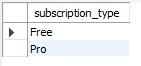
1. Enable google sheets API & download credentials as JSON
2. Setup Python environment with libraries
3. Create a data pipeline
4. Create a date structure transformation function because in MYSQL the date format is (yyyy-mm-dd) which is different from sheets
5. Create a scheduler for automatically fetch data with a certain period of interval

**EDA**

1. Find out all the unique types of subscriptions.

↪ SELECT DISTINCT subscription\_type

FROM users;



1. Find out all the unique countries.

↪ SELECT DISTINCT country

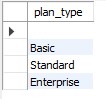
FROM users;



1. Find out all the unique types of plans.

↪ SELECT DISTINCT country

FROM users;



1. Check for duplicates.

↪ SELECT DISTINCT COUNT(user\_id)

FROM users;



1. Check if “Pro Upgrade Date” and “Plan Type” has only for Pro Users

↪ SELECT user\_id, subscription\_type, pro\_upgrade\_date, plan\_type

FROM users

WHERE subscription\_type != 'Pro'

AND pro\_upgrade\_date IS NOT NULL

AND plan\_type IS NOT NULL;



1. Check if neumerical attributes have proper values

↪ SELECT

MIN(total\_sessions) AS min\_sessions, MAX(total\_sessions) AS max\_sessions, AVG(total\_sessions) AS avg\_sessions,

MIN(page\_views) AS min\_page\_views, MAX(page\_views) AS max\_page\_views, AVG(page\_views) AS avg\_page\_views,

MIN(monthly\_revenue) AS min\_revenue, MAX(monthly\_revenue) AS max\_revenue, AVG(monthly\_revenue) AS avg\_revenue,

MIN(days\_active) AS min\_days\_active, MAX(days\_active) AS max\_days\_active, AVG(days\_active) AS avg\_days\_active

FROM users;

1. Verify date format (yyyy-mm-dd)

↪ SELECT user\_id, install\_date, last\_active\_date, pro\_upgrade\_date

FROM users

WHERE install\_date NOT REGEXP '^[0-9]{4}-[0-9]{2}-[0-9]{2}$';

1. Verification of churned data (appropriate for real-time data updates)

↪ SELECT user\_id, last\_active\_date, churned

FROM users

WHERE churned = 1 AND last\_active\_date > DATE\_SUB(CURDATE(), INTERVAL 30 DAY);