Firas Heib Portfoilo

iTunes Data Warehouse and Dashboard Project

Department Budget Integration

In the **Dim_Employee.sql** model, we integrated department budget data with employee data to provide insights into the relationship between employees and department budgets.

Key Actions:

Joining Department Data:

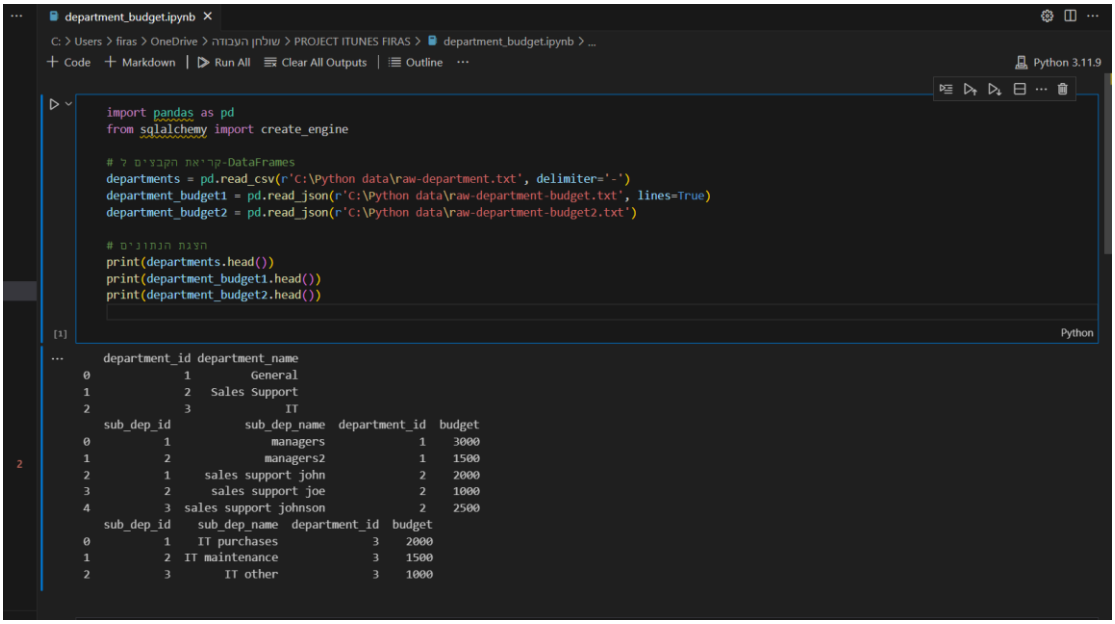
The department table included department_id, department name, and budget.

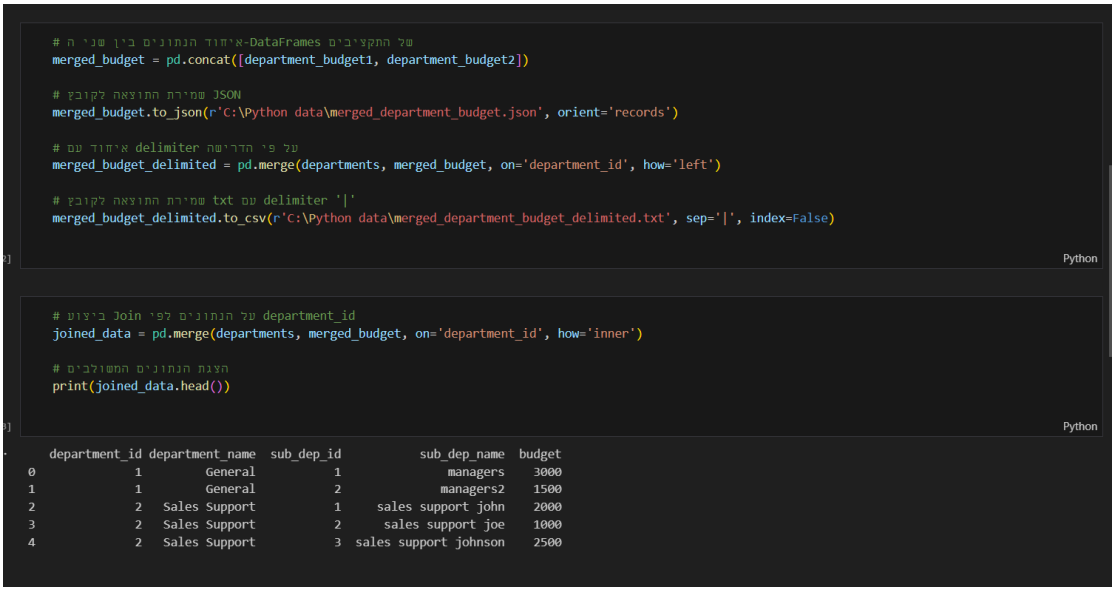
This table was joined with the employee table using the department_id field.

Calculating Department Budgets:

Each employee was assigned the budget of their respective department.

This allowed budget details to be included in the employee dimension table.





Dimensional Models (DBT) .1 ##

Dim_Customer.sql ###

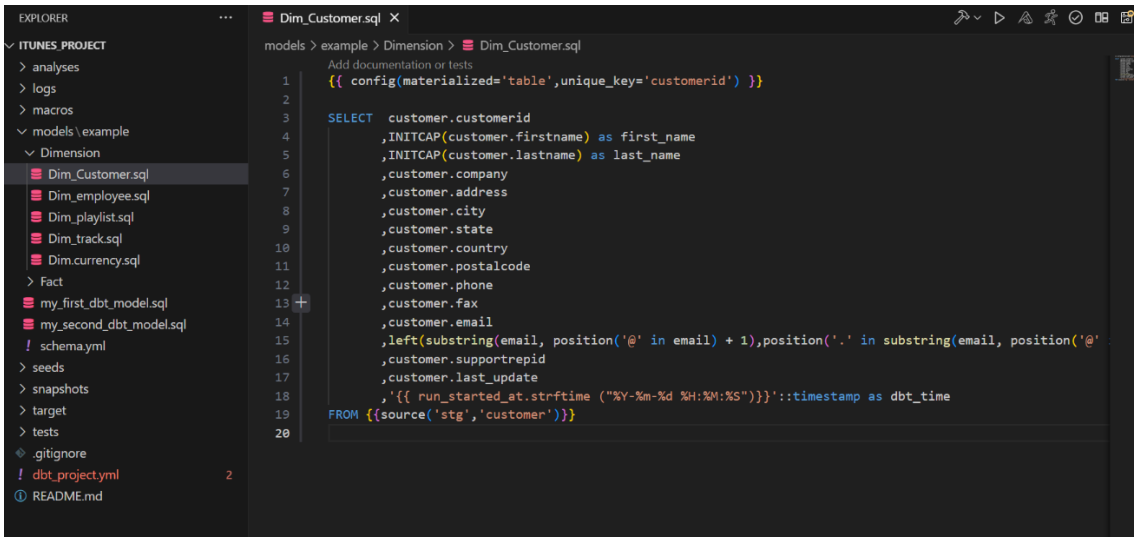
Goal:** Create a Customer Dimension table with customer details such as ** -
.name, address, phone number, and email domain extraction

:Key Actions -

.Standardized customer names to title case -

.Extracted email domains -

.Added a timestamp column (`dbt_time`) for process tracking -



Dim_Employee.sql ###

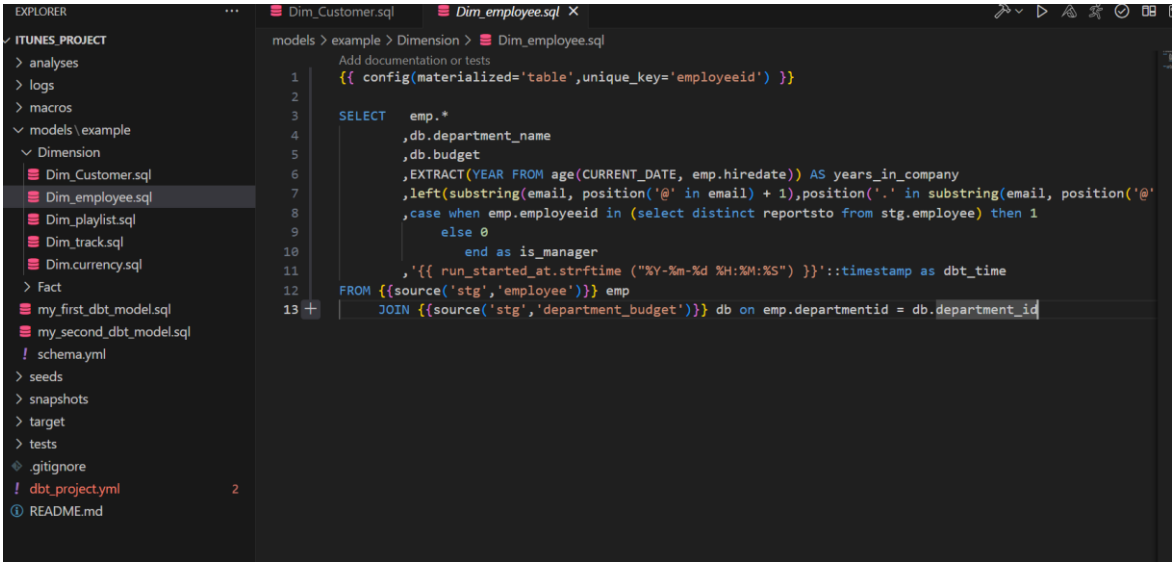
Goal:** Create an Employee Dimension table, including department budgets ** -
.and identifying managers

:Key Actions -

.Calculated years of employment for each employee -

.Identified managers using a case condition -

.Merged department budget data with employee details -



Dim_Playlist.sql ###

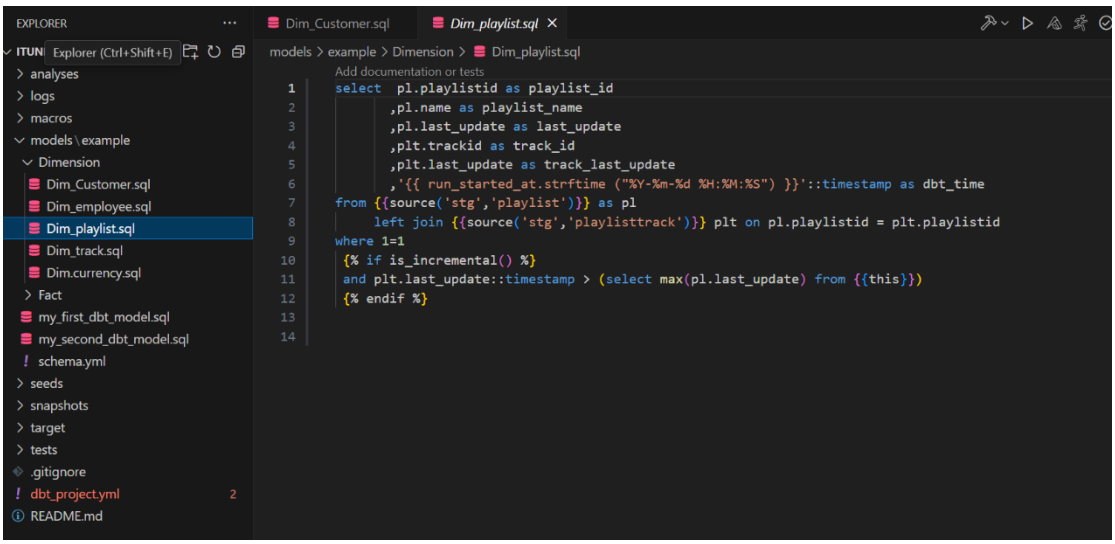
Goal:** Create a Playlist Dimension table to analyze playlists and track ** -

.updates

:Key Actions -

.Combined playlist and track details using a join -

.Implemented incremental updates with a timestamp condition -



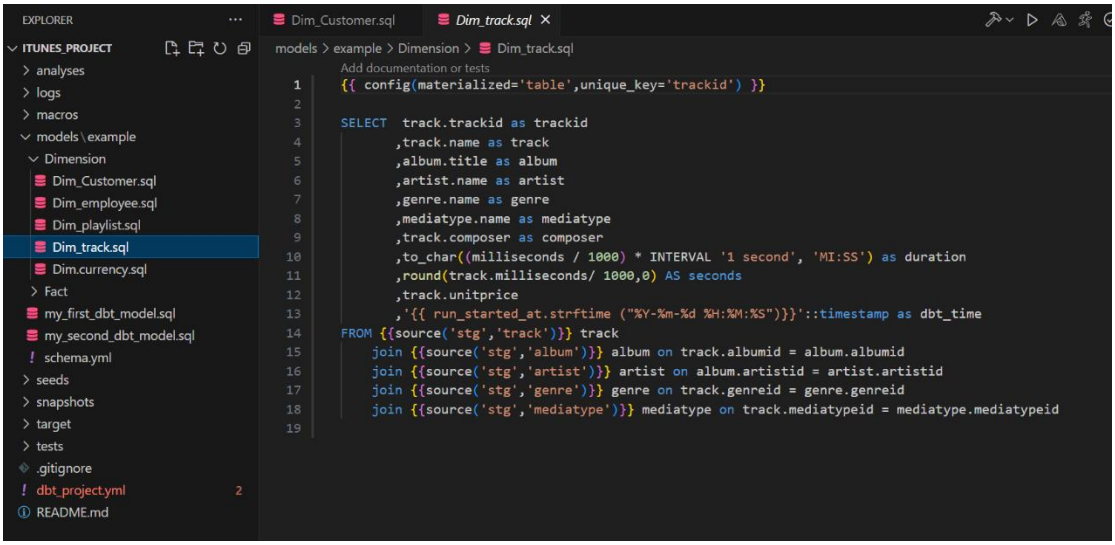
Dim_Track.sql ###

Goal:** Create a Track Dimension table with details like genre, album, artist, ** -
.track duration, and price

:Key Actions -

.Converted track duration from milliseconds to minutes:seconds format -

.Merged data from genre, album, and media type tables -



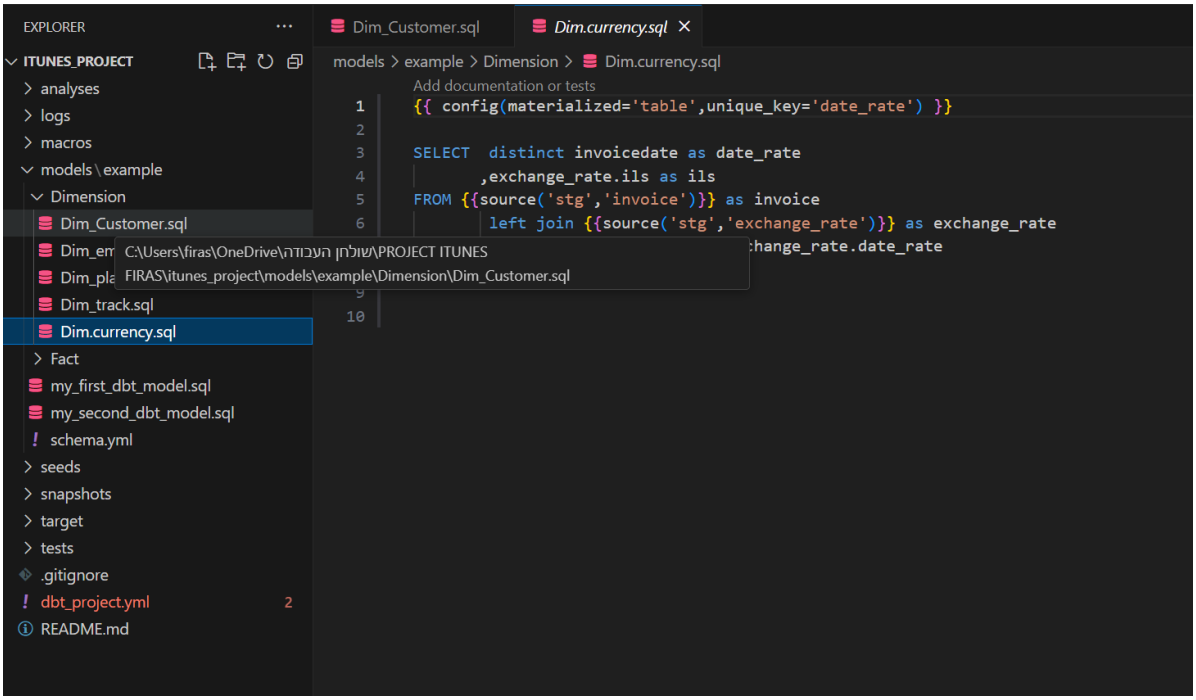
Dim_Currency.sql ###

Goal:** Build a Currency Dimension table to incorporate historical exchange ** -
.rates between USD and ILS

:Key Actions -

.Integrated exchange rates with sales data based on date -

.Prepared data for currency-based sales analysis -



Fact Models .2 ##

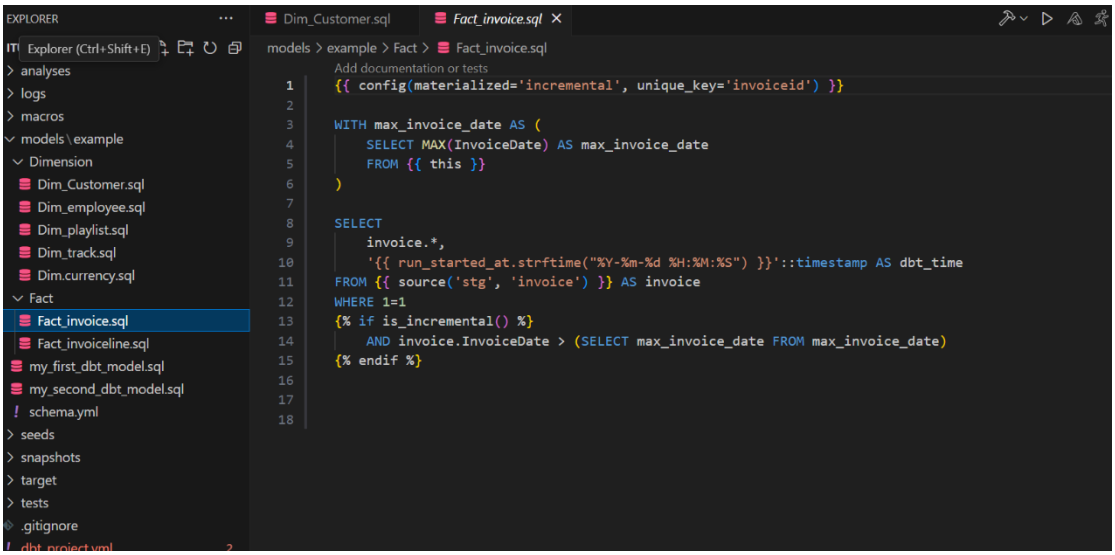
Fact_Invoice.sql ###

.Goal:** Create an Invoice Fact table for high-level sales transactions** -

:Key Actions -

.Implemented incremental updates based on the latest invoice date -

.Added process timestamp (`dbt_time`) -



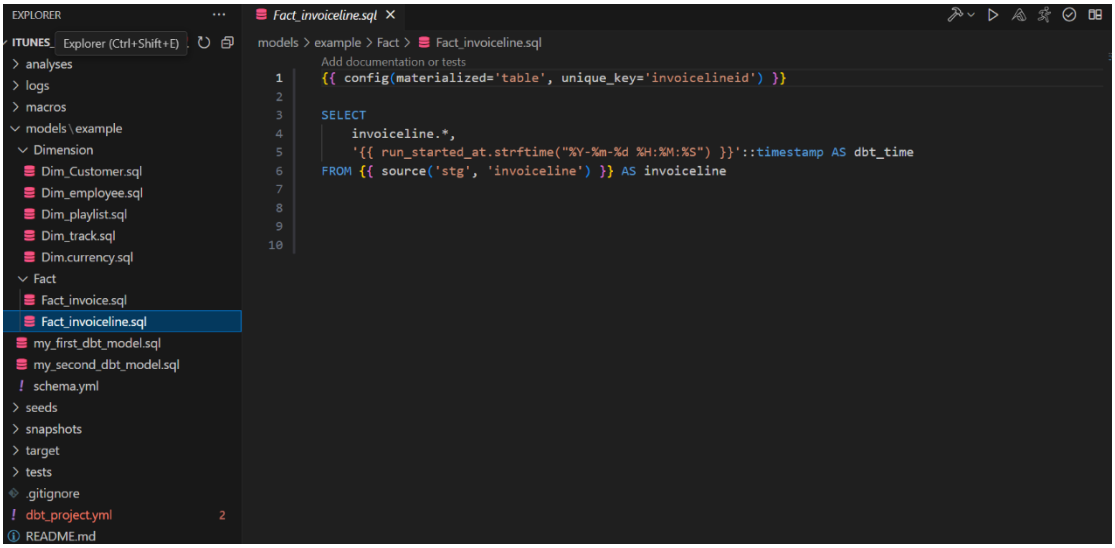
Fact_InvoiceLine.sql ###

Goal:** Create an Invoice Line Fact table for detailed sales transaction ** -
.analysis

:Key Actions -

.Stored individual line items of invoices, including quantities and prices -

.Added a timestamp column for tracking updates -



API Integration for Currency Conversion .3 ##

API_Currency.ipynb ###

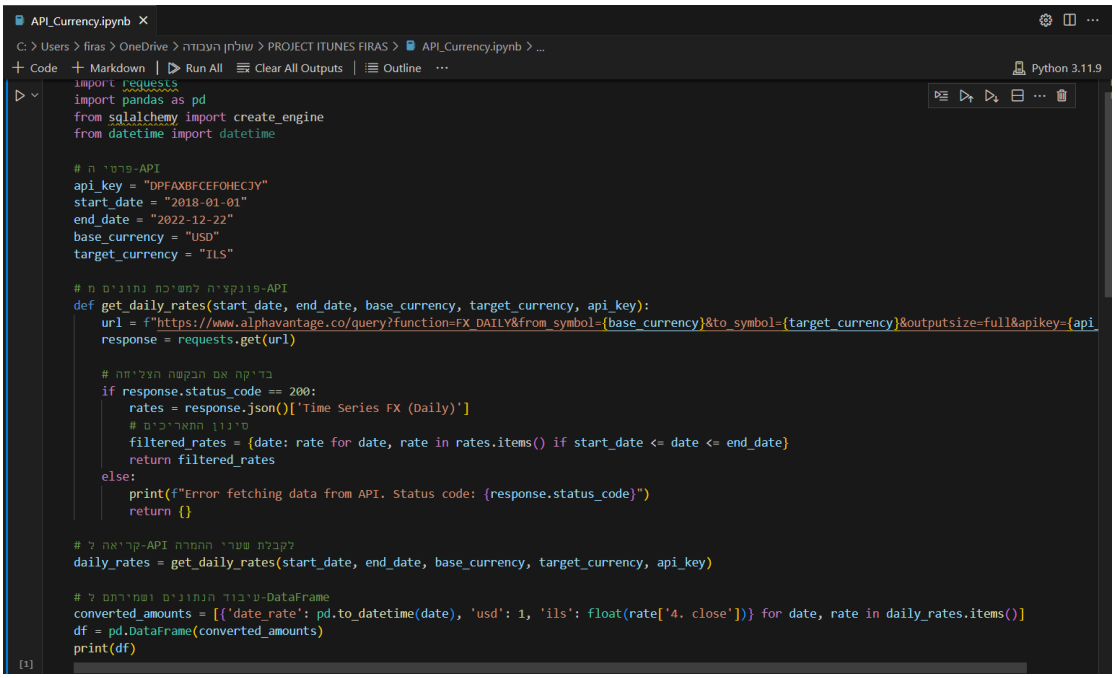
Goal:** Fetch and integrate exchange rates between USD and ILS using an ** -
.external API

:Key Actions -

.Retrieved historical exchange rates (2018-2022) via the Alpha Vantage API -

.Stored data in a PostgreSQL database as a staging table -

.Prepared data for use in the `Dim_Currency` model -



```
import requests
import pandas as pd
from sqlalchemy import create_engine
from datetime import datetime

# פרטי ה-API
api_key = "DPFAXBFCEFOHECJY"
start_date = "2018-01-01"
end_date = "2022-12-22"
base_currency = "USD"
target_currency = "ILS"

# פונקציה למשיכת נתונים מ-API
def get_daily_rates(start_date, end_date, base_currency, target_currency, api_key):
    url = f"https://www.alphavantage.co/query?function=FX_DAILY&from_symbol={base_currency}&to_symbol={target_currency}&outputsize=full&apikey={api_
    response = requests.get(url)

    # בדיקה אם הבקשה הצליחה
    if response.status_code == 200:
        rates = response.json()['Time Series FX (Daily)']
        # סינון התאריכים
        filtered_rates = {date: rate for date, rate in rates.items() if start_date <= date <= end_date}
        return filtered_rates
    else:
        print(f"Error fetching data from API. Status code: {response.status_code}")
        return {}

# קריאה ל-API לקבלת שערי ההמרה
daily_rates = get_daily_rates(start_date, end_date, base_currency, target_currency, api_key)

# עיבוד הנתונים ושמירתם ל-DataFrame
converted_amounts = [{'date_rate': pd.to_datetime(date), 'usd': 1, 'ils': float(rate['4. close'])} for date, rate in daily_rates.items()]
df = pd.DataFrame(converted_amounts)
print(df)
```

```
      date_rate  usd     ils
0    2022-12-22    1  3.4902
1    2022-12-21    1  3.4630
2    2022-12-20    1  3.4626
3    2022-12-19    1  3.4416
4    2022-12-16    1  3.4580
...         ...  ...     ...
1294 2018-01-05    1  3.4281
1295 2018-01-04    1  3.4436
1296 2018-01-03    1  3.4481
1297 2018-01-02    1  3.4475
1298 2018-01-01    1  3.4666

[1299 rows x 3 columns]
```

```
# יצירת חיבור ל-PostgreSQL (שנה את פרטי ההתחברות בהתאם למסד הנתונים שלך)
engine = create_engine('postgresql://postgres:postgres@localhost/chinook')

# טעינת הנתונים לטבלת 'exchange_rate' בסכמה 'stg'
df.to_sql('dim_currency', con=engine, if_exists='append', index=False, schema='stg')

print("Data loaded successfully into 'exchange_rate' table.")
```

Python

```
Data loaded successfully into 'exchange_rate' table.
```

Business Intelligence (Power BI Dashboards) .4 ##

Dashboard 1: Sales Overview ###

:Highlights -

Top 5 Genres:** Displayed the most popular music genres based on track ** -
.and album counts

.Top 5 Countries by Sales:** Analyzed sales by geographic location** -

.Customer Spending:** Identified top customers based on total purchases** -

.Tooltip functionality added for genre-specific insights -

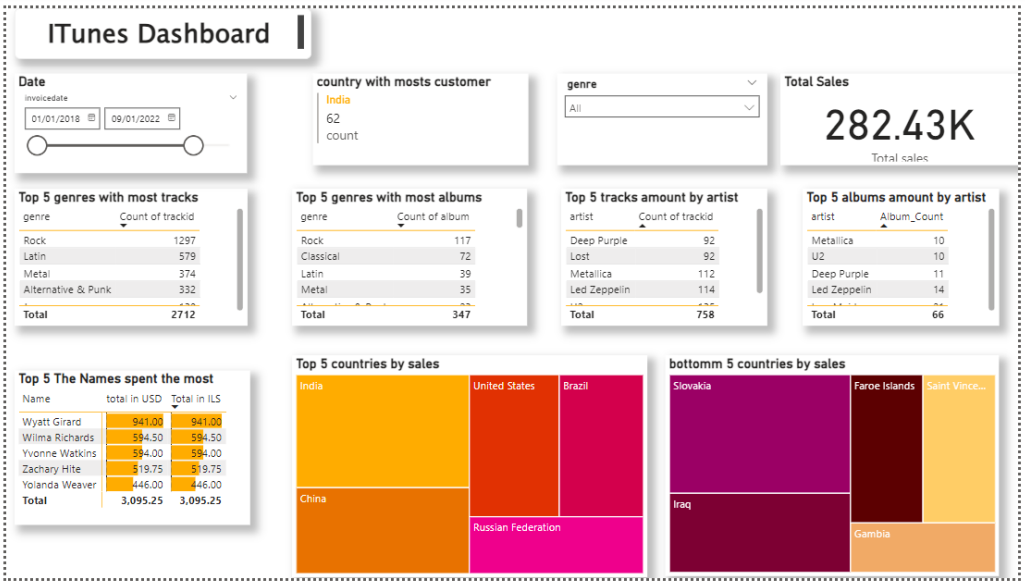
Dashboard 2: Track Analysis ###

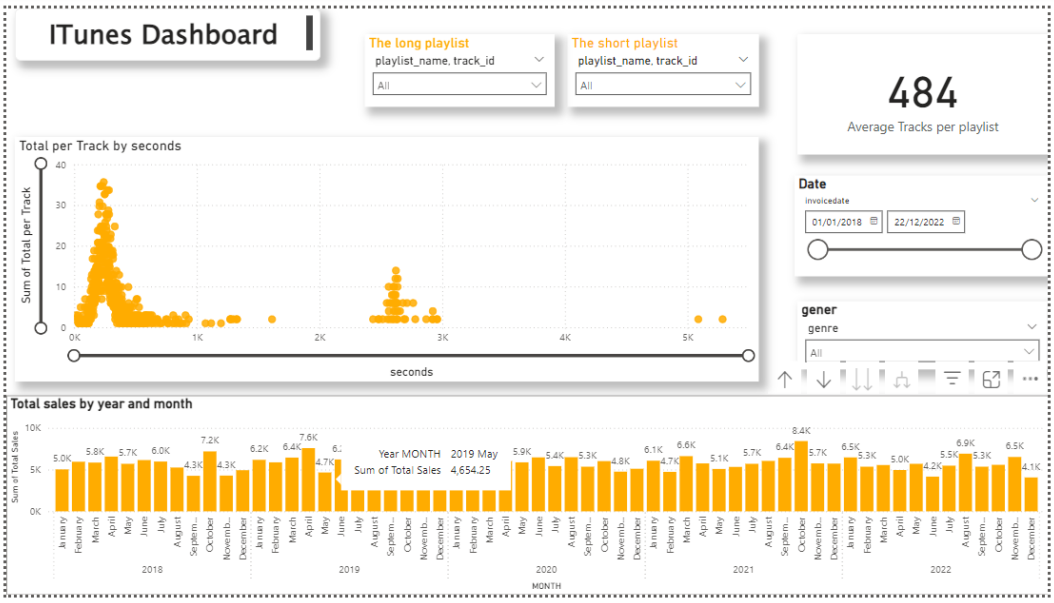
:Highlights -

.Track Duration:** Visualized distribution of track lengths in seconds** -

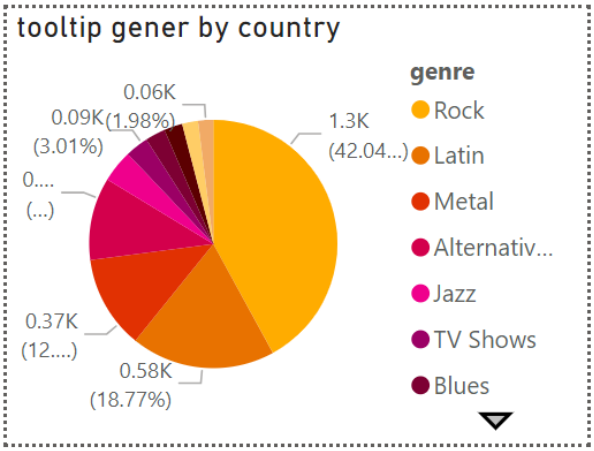
Monthly Sales Trends:** Analyzed sales patterns over time (monthly ** -
.granularity)

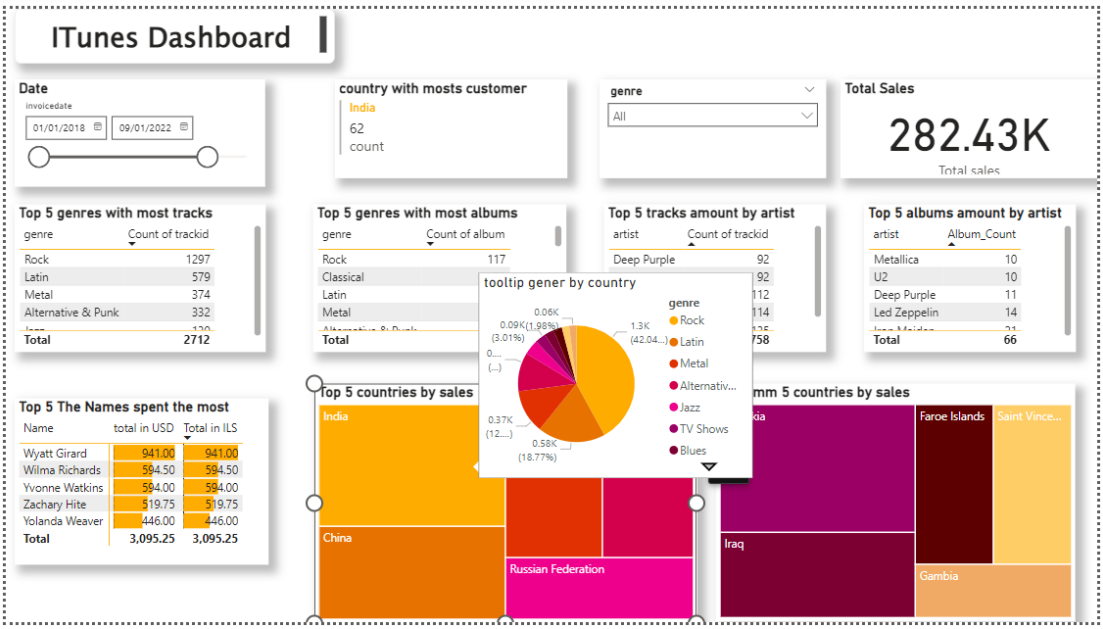
.Added slicers for dynamic filtering by date and genre -


ITunes Dashboard
Date
invoicedate
01/01/2018
09/01/2022
country with mosts customer
India
62
count
genre
All
Total Sales
282.43K
Total sales
Top 5 genres with most tracks
genre
Count of trackid
Rock 1297
Latin 579
Metal 374
Alternative & Punk 332
Total 2712
Top 5 genres with most albums
genre
Count of album
Rock 117
Classical 72
Latin 39
Metal 35
Total 347
Top 5 tracks amount by artist
artist
Count of trackid
Deep Purple 92
Lost 92
Metallica 112
Led Zeppelin 114
Total 758
Top 5 albums amount by artist
artist
Album_Count
Metallica 10
U2 10
Deep Purple 11
Led Zeppelin 14
Total 66
Top 5 The Names spent the most
Name
total in USD
Total in ILS
Wyatt Girard 941.00 941.00
Wilma Richards 594.50 594.50
Yvonne Watkins 594.00 594.00
Zachary Hite 519.75 519.75
Yolanda Weaver 446.00 446.00
Total 3,095.25 3,095.25
Top 5 countries by sales
India
United States
Brazil
China
Russian Federation
bottomm 5 countries by sales
Slovakia
Faroe Islands
Saint Vince...
Iraq
Gambia


ITunes Dashboard
The long playlist
playlist_name, track_id
All
The short playlist
playlist_name, track_id
All
484
Average Tracks per playlist
Total per Track by seconds
Sum of Total per Track
seconds
Date
invoicedate
01/01/2018
22/12/2022
gener
genre
All
Total sales by year and month
Sum of Total Sales
Year MONTH 2019 May
Sum of Total Sales 4,654.25
2018
2019
2020
2021
2022
MONTH

TOOLTIP:





Summary ##

:This project showcases end-to-end ETL and analytics capabilities

Data Engineering:** Building a Star Schema with DBT, incorporating ** -
.Dimensional and Fact tables

API Integration:** Fetching external data for currency conversion and storing ** -
.it in PostgreSQL

Data Visualization:** Developing interactive dashboards in Power BI to ** -
.provide actionable insights