

Milestone: Project

A Case Analysis on A&S Consulting Group Database Management System

Vishesh Gupta

Graduate Student, MS in Data Analytics Engineering, Northeastern University

gupta.vishe@northeastern.edu

Signature of Student: *Vishesh Gupta*

Submission Date: 10th December, 2022

I. INTRODUCTION

Background:

A&S Consulting Group (American and Spainian Consulting Group), earlier known as AS Management & Research (ASMR), is an American multinational based in Boston, USA. A&S Consulting group is involved in finance and tourism, and the company's founders are Vishesh Gupta and Amitoj Singh Kohli.

The company was established in 1999 and aimed to be one of Fortune's 500 companies. A&S Consulting Group has been gathering and playing with the data to gain and share significant insights with their esteemed customers on investment opportunities worldwide. Besides, A&S Consulting's secondary revenue generates from tourism, which feeds a country-specific knowledge base to its prospects and customers and encourages them to travel.

Business Problem:

A&S Consulting group requires customers to enroll for their services to gain deeper insights into investment opportunities. A customer can avail any of the services termed as "Match-Making Investment" (M-MI) and "Match-Making Tourism" (M-MT).

Match-Making Investment (M-MI) service stores the interested customer's information. It then provides hindsight to subscribed customers regarding the investment opportunities in a specific country based on various financial factors like GDP, inflation rate, gross savings, etc. Besides the economic factors, M&S also considers other general and miscellaneous factors. The general factors can be categorized as population and income group. In contrast, the other factors include the form of government, geo-political situation, language barriers, etc. Besides, each country has its own rules and regulations.

Match-Making Tourism (M-MT) is a cultural, economic, and social phenomenon involving people traveling to locations outside their usual environment for leisure, work, or other reasons. Likewise, the M-MT service stores the subscriber details. It then feeds them with country-specific details on the country's culture, history, places of interest, food, climatic conditions, and language with tie-ups with local guides to create a travel plan and itinerary as per their destination choice as a packaged fee.

Besides, A&S Consulting Group plans to collect feedback about its services from the customers who have availed their services to determine its strengths and weakness and improve the services.

Requirements:

Identify key entities and attributes related to this use case to gather the actual datasets / create sample datasets with adequate size to cover the business problem/need defined above. Also, draw relationships amongst identified entities and attributes and arrive at a database model/design iteratively for the logical data models. Besides, you need to design queries using an application to determine the decision variables. Further, you need to perform SQL connectivity in Python and perform analysis related to the database model.

The foundation must be built to scale up for future demands and expand the data model's coverage for commercial usage:

- 1) Country wise tie-up with local operators for ticketing and booking for travel and stay.
- 2) Provide foreign currency exchange services.

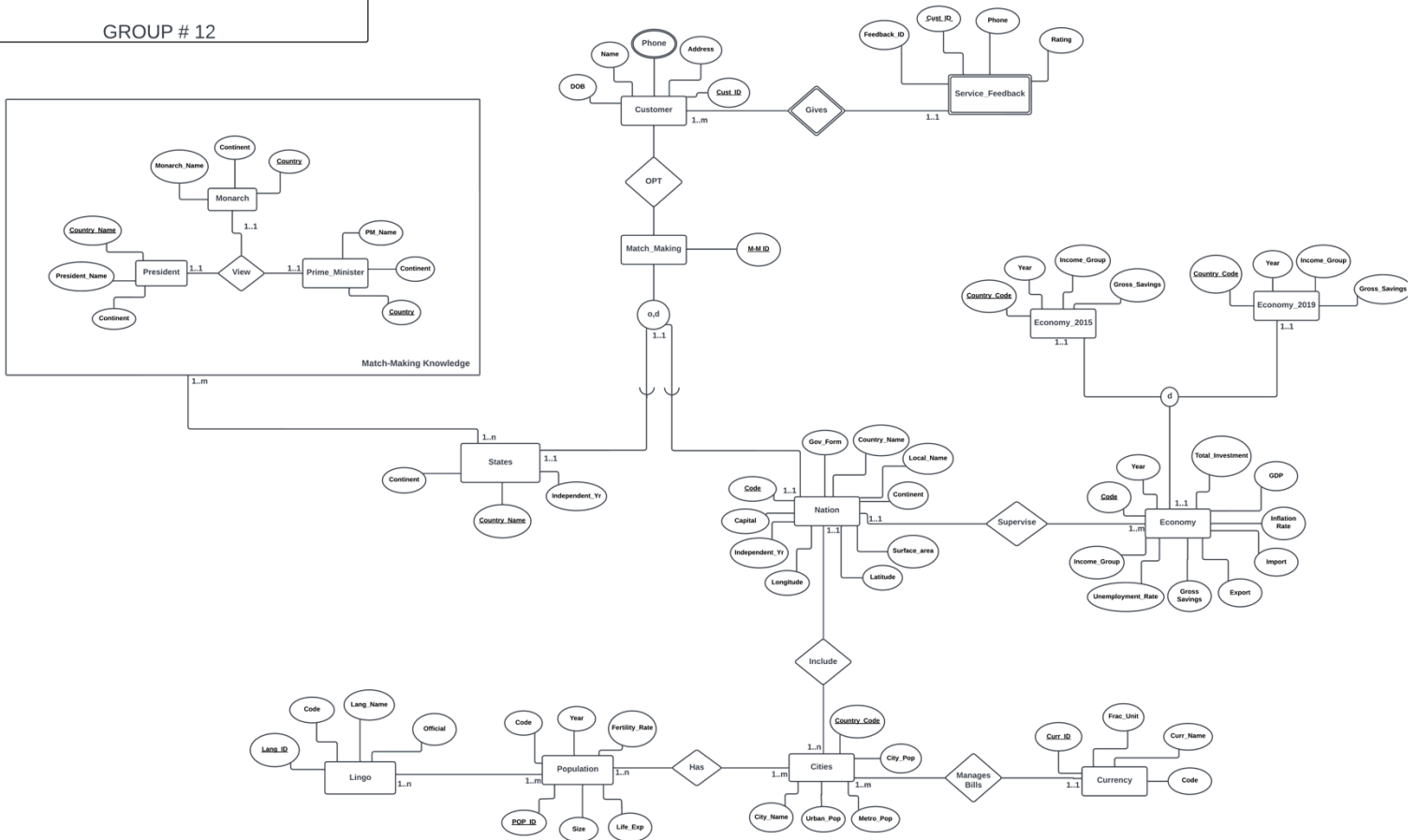
II. CONCEPTUAL DATA MODELLING

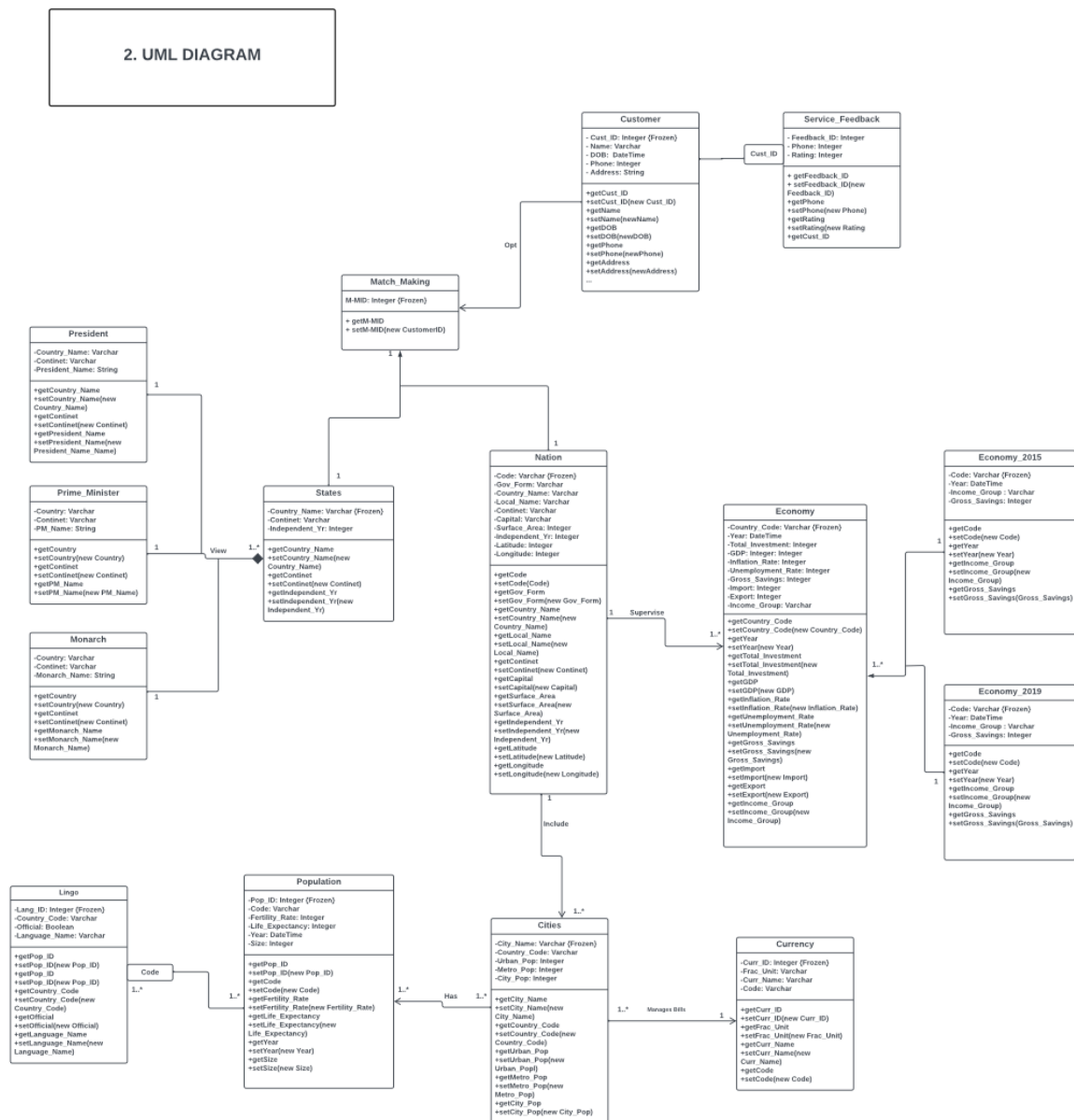
1. EER Diagram:

II Conceptual Data Modeling

1. EER DIAGRAM

GROUP # 12





III. CONCEPTUAL DATA MODELLING

1. Customer(Cust_ID , DOB, Name, Address)
Cust_ID: Primary Key, Null is not Permitted
2. Phone(C_phone, Cust_ID)
Cust_ID: Foreign key refers to the Cust_ID from Customer Relation, Null is not Permitted

3. Service Feedback(Feedback_ID, Phone, Rating, *Cust_ID*)
Cust_ID: Foreign key refers to the Cust_ID from Customer Relation, Null is not Permitted
4. Match-Making(M-MID, *Cust_ID*)
M-MID: Primary Key, Null is not Permitted
Cust_ID: Foreign key refers to the Cust_ID from Customer Relation, Null is not Permitted
5. Nation(N_Code, Gov_Form, Country_Name, Local_Name, Continent, Surface_Area, Capital, Independent_YR, Latitude, Longitude)
Code: Primary Key, Null is not Permitted
6. Economy(Code, Year, Income_Group, Total_Investment, GDP, Inflation_Rate, Unemployment_Rate, Gross_Savings, Import, Export, *N_Code*)
Code: Primary Key, Null is not Permitted
7. Supervise(*N_Code*, *Code*)
N_Code: Foreign key refers to the N_Code from Nation Relation, Null is not Permitted
Code: Foreign key refers to the Code from Economy Relation, Null is not Permitted
8. Economy_2015(Country_Code, Year, Income_Group, Gross_Savings, *Code*)
Country_Code: Primary Key, Null is not Permitted
Code: Foreign key refers to the Code from Economy Relation, Null is not Permitted
9. Economy_2019(Country_Code, Year, Income_Group, Gross_Savings, *Code*)
Country_Code: Primary Key, Null is not Permitted
Code: Foreign key refers to the Code from Economy Relation, Null is not Permitted
10. Cities(Country_Code, City_Pop, Metro_Pop, Urban_Pop, City_Name, *N_Code*)
Country_Code: Primary Key, Null is not Permitted
N_Code: Foreign key refers to the N_Code from Nation Relation, Null is not Permitted
11. Currencies(Curr_ID, Frac_Unit, Curr_Name, *Code*)
Curr_ID: Primary Key, Null is not Permitted
12. Population(Pop_ID, Code, Year, Fertility_Rate, Life_Exp, Size, *Country_Code*)
Pop_ID: Primary Key, Null is not Permitted
Country_Code: Foreign key refers to the Country_Code from Cities Relation, Null is not Permitted
13. Lingo(Lang_ID, Code, Lang_Name, Official)
Lang_ID: Primary Key, Null is not Permitted

14. States(Country_Name, Continent, Independent_YR, *M-MID*)
Country_Name: Primary Key, Null is not Permitted
M-MID: Foreign key refers to the M-MID from Match-Making Relation, Null is not Permitted
15. President(Country_Name, Continent, President_Name)
Country_Name: Primary Key, Null is not Permitted
16. Prime_Minister(Country, Continent, PM_Name)
Country: Primary Key, Null is not Permitted
17. Monarch(Country, Continent, Monarch_Name)
Country: Primary Key, Null is not Permitted

IV. IMPLEMENTATION IN MySQL

Q1) Display City name, country code, Prime Minister name, Country name, City Population, city Urban Population, Surface Area of country, country constitution name, and its currency name and its fractional unit which has a metro population not equals to zero and display 12 records for only India, United Kingdom, oman, Australia, Pakistan countries and sort the metro population column in ascending order.

Select distinct c.name_city as City_Name, c.country_code as Country_Code, pm.prime_minister as Prime_Ministers, n.country_name as Country_Name, c.city_proper_pop as City_Pop, c.metroarea_pop as Metro_Pop, c.urbanarea_pop as Urban_Pop, n.surface_area as Surface_Area_C, n.gov_form as Constitution_Name, ps.curr_code as Currency_Code, ps.frac_unit **from** cities as c **inner join** nation as n **on** c.country_code = n.code **inner join** currency as ps **on** n.code = ps.code **inner join** prime_ministers as pm **on** n.country_name = pm.country **where** metroarea_Pop > (select min(Metroarea_Pop) **from** cities **WHERE** cities.country_code = ps.code) **and** country_code IN ('IND', 'GBR', 'OMN', 'AUS', 'PAK') **order by** 6 limit 12;

City_Name	Country_Code	Prime_Ministers	Country_Name	City_Pop	Metro_Pop	Urban_Pop	Surface_Area_C	Constitution_Na...	Currency_Code	frac_unit
Bhopal	IND	Narendra Modi	India	1798218	1864389	1798218	3287260	Federal Republic	INR	Paisa
Islamabad	PAK	Shehbaz Sharif	Pakistan	1900000	2200000	1900000	796095	Republic	PKR	Paisa
Patna	IND	Narendra Modi	India	1683200	2231554	1683200	3287260	Federal Republic	INR	Paisa
Nagpur	IND	Narendra Modi	India	2405665	2497870	2405665	3287260	Federal Republic	INR	Paisa
Kanpur	IND	Narendra Modi	India	2768057	3152317	2768057	3287260	Federal Republic	INR	Paisa
Faisalabad	PAK	Shehbaz Sharif	Pakistan	6480765	3675000	6480765	796095	Republic	PKR	Paisa
Kochi	IND	Narendra Modi	India	2232456	4221140	2232456	3287260	Federal Republic	INR	Paisa
Visakhapatnam	IND	Narendra Modi	India	2035922	5340000	2035922	3287260	Federal Republic	INR	Paisa
Bengaluru	IND	Narendra Modi	India	8425970	9807000	8425970	3287260	Federal Republic	INR	Paisa
Lahore	PAK	Shehbaz Sharif	Pakistan	10355000	13569000	10355000	796095	Republic	PKR	Paisa
London	GBR	Boris Johnson	United Kingdom	8673713	13879757	8673713	242900	Constitutional Mo...	GBP	Penny
Kolkata	IND	Narendra Modi	India	4486679	14667000	4486679	3287260	Federal Republic	INR	Paisa

Q2) Retrieve several details for the Year 2010 population that indicate whether the official language of a particular country(code) is True or False, as well as the percentage of each language spoken in the country and the life expectancy for 2010 with the country code.

with Invest_CTE as (select country_code, year, pop_id, life_expectancy, size **from** population as p **where** p.year in (select year **from** population **where** year = '2010') **and** life_expectancy > 58 **and** life_expectancy is not null)

```

select l.lang_id as ID, l.name as lang_name, l.code as country_code, l.official as
Official_language, l.percent, Invest_CTE.life_expectancy, Invest_CTE.year
from lingo as l Right JOIN Invest_CTE
on l.code = Invest_CTE.country_code
where percent > (select min(percent) from lingo)
order by l.lang_id asc;

```

	ID	lang_name	country_code	Official_langua...	percent	life_expectan...	year
▶	1	Dari	AFG	TRUE	50	58.97082927	2010
	2	Pashto	AFG	TRUE	35	58.97082927	2010
	3	Turkic	AFG	FALSE	11	58.97082927	2010
	4	Other	AFG	FALSE	4	58.97082927	2010
	5	Albanian	ALB	TRUE	98.8	77.03695122	2010
	6	Greek	ALB	FALSE	0.5	77.03695122	2010
	7	Other	ALB	FALSE	0.6	77.03695122	2010
	8	unspecified	ALB	FALSE	0.1	77.03695122	2010

Q3) Display Country code, income groups containing the word "lower" or "low" and gross savings exceeding 20 for the economical year 2015

```

select distinct e5.code, e5.income_group, e5.gross_savings from economy_2015 as e5
where income_group IN (select income_group from economy_2015 where
(income_group like "%Lower" or income_group like "%Low%")) and gross_savings > 20;

```

code	income_group	gross_savings
COG	Lower middle income	40.45683662
CPV	Lower middle income	28.15598428
DJI	Lower middle income	25.67934937
DZA	Lower middle income	36.48792155
ETH	Low income	29.52215852
GHA	Lower middle income	21.22924497
IDN	Lower middle income	30.12335649
IND	Lower middle income	32.4539249
KHM	Lower middle income	20.72817128

Q4) Find the country and continent whose independence year is between 1910 and 1970, and whose economic behavior is characterized by an inflation rate and GDP percapita that are greater than their own averages for the year 2010.

```

SELECT country_name, continent, Indep_Year, Region, e.inflation_rate from
(select Code, Country_name, Continent, Region, Indep_Year from nation where indep_year
between '1910' and '1970') as sub1 inner join economy as e on e.code = sub1.code where
inflation_rate > ( select avg(inflation_rate) from economy where gdp_percapita > (select
avg(gdp_percapita) from economy)) and year = '2010';

```

Country_name	Continent	Indep_Year	Region	inflation_ra...
▶ Afghanistan	Asia	1919	Southern and Central Asia	2.179
Albania	Europe	1912	Southern Europe	3.605
Algeria	Africa	1962	N or thern Africa	3.913
Barbados	N or th America	1966	Caribbean	5.824
Benin	Africa	1960	Western Africa	2.179
Bhutan	Asia	1910	Southern and Central Asia	5.726
Botswana	Africa	1966	Southern Africa	6.95
Burundi	Africa	1962	Eastern Africa	6.496

Q5) Retrieve 15 feedback Id's and the corresponding rating given by different customers for the services offered to them along with their first and last name.

```

SELECT Feedback_Id, Customer_First_Name, Customer_Last_Name, Rating FROM feedback
JOIN customer ON feedback.Cust_Id=customer.Cust_Id
WHERE rating IN ('Excellent', 'Good')
ORDER BY 4 LIMIT 15;

```

	Feedback_Id	Customer_First_Na...	Customer_Last_Na...	Rating
	3	George	Donald	Excellent
	12	Sam	Li	Excellent
	1	Jennifer	Rose	Excellent
	10	Manuel	Ola	Excellent
	13	Darren	Glen	Excellent
	2	Anita	Mehlotra	Excellent
	4	Amit	Sinoh	Excellent

Q6) Write a SQL query using EXISTS to retrieve the details of only those customers who have given an 'excellent' rating along with customer name and customer country.

```
SELECT customer_first_name AS First_Name, customer_last_name AS Last_Name, country
FROM customer c
WHERE EXISTS (SELECT *
FROM service_feedback AS sf
WHERE sf.rating = 'Excellent'
AND c.cust_id = sf.cust_id);
```

Result Grid			Filter Rows: <input type="text" value="Search"/>	Export:
First_Name	Last_Name	country		
Harry	Chris	Australia		
Darren	Glen	Australia		
Anita	Mehlotra	Canada		
Sam	Li	China		
Ben	Cook	GBR		
Vish	Kumar	India		
Amit	Singh	India		
Finn	Noah	New Zealand		
Manuel	Ola	Singapore		
Andrew	John	South Africa		
Jennifer	Rose	Spain		
Kevin	Wright	Switzerland		
George	Donald	USA		

Q7) Find the Investment Prediction of all the countries for making investment by investor based on the year of 2015 and generate those countries name and their sub attributes which has greater than or equal to maximum gdp_per capita.

```
select en.code as Code_Country, nt.country_name as CN_Name, en.year as Investment_Year,
en.gross_savings as Gross_Sav ,cn.basic_unit as currency_name, (en.imports + en.exports) as
Investment_Trade, case when en.imports > en.exports then 'Investment for Trading is Bad'
when en.imports < en.exports then 'Investment for Trading is Good' else '50-50 Chances for
Investment' end as Prediction from economy as en left join currency as cn on en.code = cn.code
left join nation as nt on en.code = nt.code where en.year = '2015' and en.gdp_per capita >=
(select max(en.gdp_per capita) from economy)and cn.basic_unit is not null order by 6 desc;
```

	Code_Country	CN_Name	Investment_Year	Gross_Sav	currency_name	Investment_Trade	Prediction	
▶	MLI	Mali	2015	10.118	West African CFA franc	61.114999999999995	Investment for Trading is Bad	
▶	IRL	Ireland	2015	31.998	Euro	56.126	Investment for Trading is Good	
▶	DJI	Djibouti	2015	19.039	Djiboutian franc	44.013999999999996	Investment for Trading is Bad	
▶	JAM	Jamaica	2015	12.111	Jamaican dollar	42.023	Investment for Trading is Good	
▶	GUY	Guyana	2015	8.214	Guyanese dollar	28.972	Investment for Trading is Bad	
▶	ETH	Ethiopia	2015	31.269	Ethiopian birr	28.922	Investment for Trading is Bad	
▶	LUX	Luxembourg	2015	24.865	Euro	26.857999999999997	Investment for Trading is Bad	
▶	VNM	Vietnam	2015	28.055	Vietnamese dong	25.413	Investment for Trading is Bad	

V. IMPLEMENTATION IN NoSQL

Q1) Display the result of updating the population collection with population id number 19 by increasing its population size by 100.

```
ASConsutingGroup> db.population.update({pop_id: 19},
{$inc:{ size: 100}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
ASConsutingGroup> █
```


Q2) Update the prime minister collection by adding the new Prime Minister of the United Kingdom, Rishi Sunak, and display the result.

```
ASConsutingGroup> db.primeminister.update({prime_minister: 'Boris Johnson'}, {$set: {'country': 'United Kingdom', 'continent': 'Europe', 'prime_minister': 'Rishi Sunak'}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
ASConsutingGroup> db.primeminister.find({prime_minister: 'Boris Johnson'})
ASConsutingGroup> db.primeminister.find({prime_minister: 'Rishi Sunak'})
[
  {
    _id: ObjectId("6384d865bfc91d9ce715c035"),
    country: 'United Kingdom',
    continent: 'Europe',
    prime_minister: 'Rishi Sunak'
  }
]
ASConsutingGroup> █
```

Aggregation Pipeline

Q3) Find a country with a government in the form of a republic, on the European continent, with a land area greater than 600,000 square feet.

```
ASConsutingGroup> db.Nation.aggregate([{"$match": {
  "$and": [{ 'gov_form': 'Republic' }, { 'continent': 'Europe' }, { 'surface_area': { '$gt': 600000 } } ] } ]])
```

```
[
  {
    _id: ObjectId("6384d14abfc91d9ce715b710"),
    code: 'UKR',
    country_name: 'Ukraine',
    continent: 'Europe',
    region: 'Eastern Europe',
    surface_area: 603700,
    indep_year: 1991,
    local_name: 'Ukrajina',
    gov_form: 'Republic',
    capital: 'Kiev',
    cap_long: 30.5038,
    cap_lat: 50.4536
  }
]
ASConsutingGroup> █
```

Q4) Show the total gross savings of each economic group.

```
ASConsutingGroup> db.Economy2015.aggregate([{"$group": {
  '_id': {'income_group': '$income_group'},
  'Total_Gross_savings': {'$sum': '$gross_savings'} } ]])
```

```
[
  {
    _id: { income_group: 'Upper middle income' },
    Total_Gross_savings: 932.522630047
  },
  {
    _id: { income_group: 'High income' },
    Total_Gross_savings: 1352.62928222
  },
  {
    _id: { income_group: 'Lower middle income' },
    Total_Gross_savings: 1216.0926530029
  },
  {
    _id: { income_group: 'Low income' },
    Total_Gross_savings: 241.70534349000002
  }
]
ASConsutingGroup> █
```

Map Reduce Pipeline

Q5) Display the Average gross savings of each economic group for the 2015 year.

```
ASConsutingGroup> db.Economy.mapReduce(
...   function() { emit(this.income_group, this.gross_savings); },
...   function(key, values) {return Array.avg(values)}, {
...     query: {year: 2015},
...     out: "avg2015_gross"
...   }
... )
{ result: 'avg2015_gross', ok: 1 }
db.avg2015_gross.find()
```

```
ASConsutingGroup> db.Economy.mapReduce(
...   function() { emit(this.income_group, this.gross_savings); },
...   function(key, values) {return Array.avg(values)}, {
...     query: {year: 2015},
...     out: "avg2015_gross"
...   }
... )
{ result: 'avg2015_gross', ok: 1 }
ASConsutingGroup> db.avg2015_gross.find()
[
  { _id: 'Low income', value: 12.350517241379313 },
  { _id: 'High income', value: 21.50370175438596 },
  { _id: 'Lower middle income', value: 15.867440000000002 },
  { _id: 'Upper middle income', value: 18.758703703703706 }
]
ASConsutingGroup> █
```

VI. DATABASE ACCESS via PYTHON

The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using `mysql.connector`, followed by `cursor.execute` to run and fetchall from query, followed by converting the list into a dataframe using `pandas` library and using `matplotlib`, `seaborn`, and `plotly` libraries to plot the graphs for the analytics.



VII. SUMMARY AND RECOMMENDATION

The A&S Consultancy Group database, designed from scratch on MySQL, is an industry-ready relational database that can be implemented in the industries involved in investment and travel portfolios. This database can provide the firm and clients with analytical and visualization insights to make better decisions. A glimpse of the analysis is shown in this report using Python. The foundation is built to scale up for future demands and expand the data model's coverage for commercial usages, like country-wise tie-up with local operators for ticketing and booking for travel and stay, besides providing foreign currency exchange services.