**Business establishment recommendation system based on country's economic freedom**

Team “Northern Lights”

Gunasekhar Vinugolu - 001586178 git hub: [gunasekharv6](https://github.com/gunasekharv6)

Ajay Sureka - 002165619,

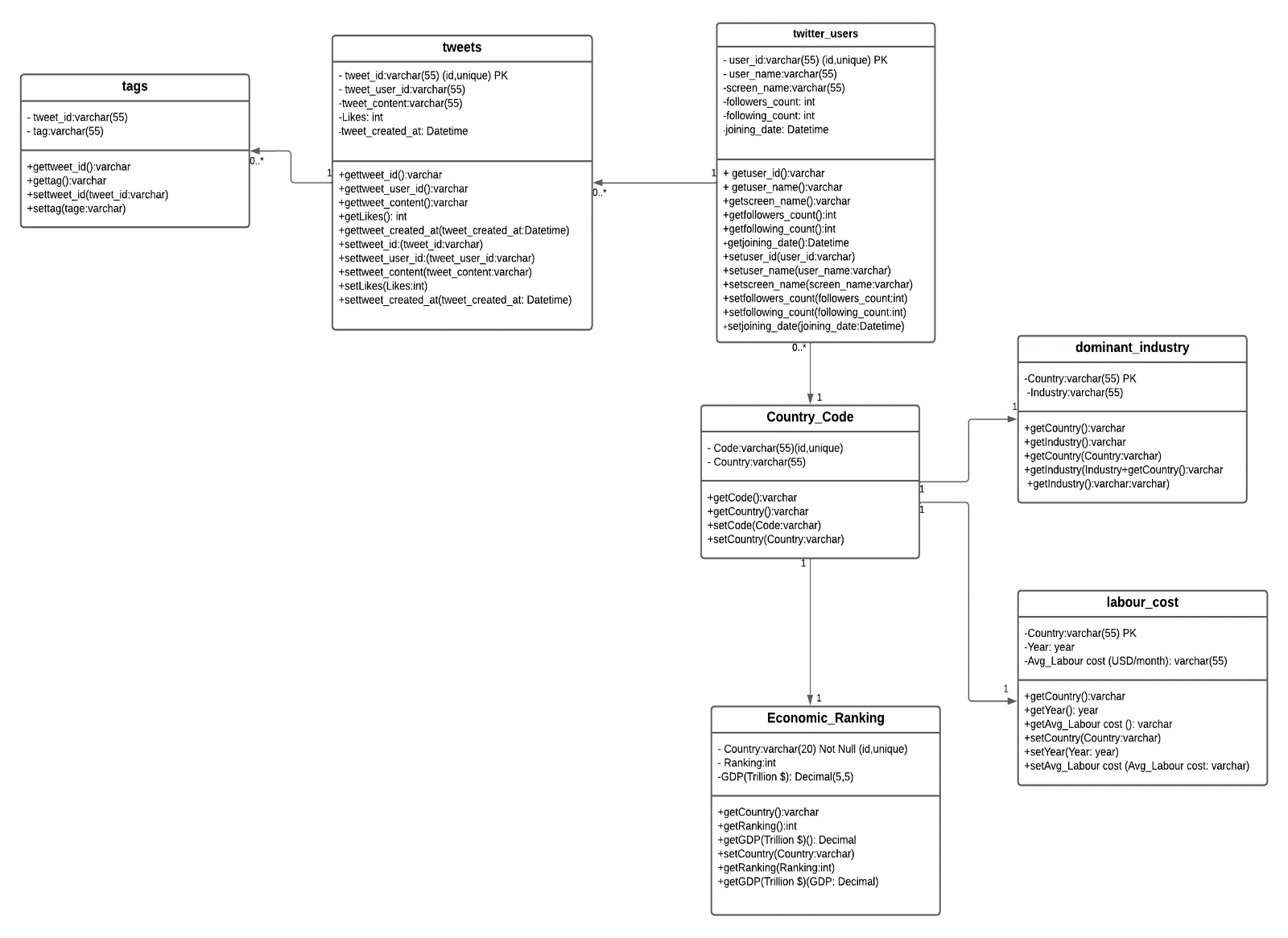
Mansi Dandgaval - 002956418,

Akhila Boppana - 001548272

Steps to run the file:

1. Install Jupyter Notebook (Anaconda 3)
2. Install MySQL Workbench
3. Download the file run it using Jupyter Notebook
4. Establish connection with the Database using MySQL Workbench –
5. Change the root username and password to MySQL workbench username and password in the code (in Jupyter Notebook)
6. Run all files using Kernel in Jupyter Notebook

ER Daigram:



**Use Cases:**

1. **Use Case:** Find the leading industries in the countries with the highest GDP and lowest labor cost respectively

**Description:** User views the countries whose largest sector in economy is Agriculture

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the countries with Agriculture sector

**System Responses:** the list of countries with Agriculture sector are displayed

**Post Condition:** system displays the list of countries for the condition

**Error:** No country available

**SQL Query:**

SELECT d.Industry, e.ranking, e.`GDP(Trillion $)`, lc.`Avg\_Labour cost (USD/month)` FROM dominant\_industry as d

INNER JOIN economic\_ranking as e

ON d.Country = e.Country

INNER JOIN labour\_cost lc

ON d.country=lc.country

WHERE e.Ranking=1;

1. **Use Case:** List of country codes with Automobile industry and their respective GDP rankings

**Description:** User views the countries whose largest sector in economy is Automobile

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the countries with Automobile industry

**System Responses:** the list of countries with Automobile are displayed

**Post Condition:** system displays the list of countries for the condition

**Error:** No country available

**Query:**

SELECT code, di.country, er.`GDP(Trillion $)` FROM country\_code cc

JOIN dominant\_industry di

ON cc.country = di.country

JOIN economic\_ranking er

ON er.country = cc.country

where di.industry LIKE "%automobile%";

1. **Use Case:** What is the Least popular tag and the country name present in that tag and that country's associated gdp

**Description:** Least popular tag is the tag with lowest number of tweets. This tag must contain a country name as part of it and what is that country's GDP.

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the least popular tag with the country name as part of that tag and that country's GDP value

**System Responses:** GDP of a country, tag name

**Post Condition:** System displays the GDP of a country for the condition

**Error:** No country available

**Query:**

select tag, Country\_In\_Tag, `GDP(Trillion $)`,tweet\_content from(

(SELECT \* FROM economic\_ranking er)D

JOIN

(

SELECT B.tag, C.tweet\_content,substring(B.tag, 2, length(tag)) as Country\_In\_Tag FROM

((SELECT \* FROM tweets tw)C

JOIN (SELECT tag, count(\*) as tagcount, tweet\_id from tags group by tag order by tagcount limit 1) B

ON B.tweet\_id = C.tweet\_id)

)E

ON E.Country\_In\_Tag = D.country

);

1. **Use Case:** Which country code has the lowest labor cost and identify the associated industry

**Description:** User views the countries whose largest sector in economy is Automobile

**Actor:** User

**Precondition:** User must have access to data and relevant tables

**Steps:**

**Actor action:** User view the country name that has lowest labour cost and dominant industries in that country

**System Responses:** Labour cost and dominant industry according to gdp

**Post Condition:** System displays the labour cost and dominant\_industry of a country for the condition

**Error:** No country available

**Query:**

SELECT D.Code, C.Country, C.`Avg\_Labour cost (USD/month)`, C.Industry FROM

((

SELECT A.Country, A.`Avg\_Labour cost (USD/month)`, B.Industry FROM

(

(SELECT \* FROM dmdd.labour\_cost)A

INNER JOIN

(SELECT \* from dmdd.dominant\_industry)B

on A.Country = B.Country

)

)C

inner join

(SELECT \* FROM dmdd.country\_code)D

on C.Country=D.Country

)

order by CAST(C.`Avg\_Labour cost (USD/month)` as float) limit 1;

1. **Use Case:** Identify the most popular tag and the associated user and that user's tweet with the most likes and which contains a GDP keyword in it

**Description:** User views the most popular tag, the one tweeted it and among that users' tweets, what is the most liked tweet and that tweet must contain the keyword "GDP"

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the most popular tag, user associated with that tag and that user's most liked tweet.

**System Responses:** tag, user\_id, tweet\_content, likes

**Post Condition:** System displays tag, most liked tweet and the user associated with both of these according to the condition

**Error:** No country available

**Query:**

(select \* from(

(SELECT tweet\_user\_id, tweet\_content, max(likes) FROM tweets) C

JOIN

(SELECT tag, tu.user\_id, tu.user\_name, tw.tweet\_id, tw.tweet\_content from

(select K.tag, tweet\_id, max(K.tot\_tags) from (select tag, tweet\_id,(count(tag)) tot\_tags from dmdd.tags group by tag ORDER BY tot\_tags DESC LIMIT 1)K)A

JOIN tweets tw

ON tw.tweet\_id = A.tweet\_id

JOIN twitter\_users tu

ON tu.user\_id = tw.tweet\_user\_id WHERE tw.tweet\_content LIKE "%GDP%")A

ON A.user\_id = C.tweet\_user\_id));

1. **Use Case:** View the country with GDP ranking 1 and find the industry of expertise along with the corresponding labor cost

**Description:** User can search for 1st ranked country in GDP, industry of expertise and labor costs

**Actors:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action** – User searches for the rankings, industry of expertise and labor costs

**System Responses** – Country which is ranked 1st in the GDP column is shown to the user

**Post Condition:** User gets the 1st ranked country in GDP

**Alternate Path:** There is no 1st Ranked country in GDP

**Error:** No rankings available

**SQL Query:**

SELECT d.Industry, e.ranking, e.`GDP(Trillion $)`, lc.`Avg\_Labour cost (USD/month)` FROM dominant\_industry as d

INNER JOIN economic\_ranking as e

ON d.Country = e.Country

INNER JOIN labour\_cost lc

ON d.country=lc.country

WHERE e.Ranking=1;

**Relational Algebra:**

πd . industry, e . ranking, e.`GDP(Trillion $)`, lc.`Avg\_Labour cost (USD/month)` σe . ranking = 1  
  (ρddominant\_industry ⋈e economic\_ranking)

(ρeCountry ⋈lc labour\_cost)

1. **Use Case:** View the industries leading in the bottom 5 countries by their GDP rankings and obtain the labor costs

**Description:** User views the countries above a particular rank in the GDP, labor costs and industries

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the countries above a particular ranking

**System Responses:** the list of countries above a rank are displayed

**Post Condition:** system displays the list of countries for the condition

**Error:** No rankings available

**Query:**

SELECT C.Country, C.Ranking,C.`GDP(Trillion $)`, C.Industry,D.`Avg\_Labour cost (USD/month)` FROM

((

SELECT A.Country, A.Ranking, A.`GDP(Trillion $)`, B.Industry FROM

(

(SELECT \* FROM dmdd.economic\_ranking)A

INNER JOIN

(SELECT \* from dmdd.dominant\_industry)B

on A.Country = B.Country

)

)C

inner join

(SELECT \* FROM dmdd.labour\_cost)D

on C.Country=D.Country

)

having Ranking>55

Order by `Avg\_Labour cost (USD/month)`;

1. **Use Case:** Find the list of country codes with labour costs less than $500/ month and the corresponding industry

**Description:** User views the countries below a particular labor cost in the Labor\_Cost table in this condition it is $500

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the countries above a particular labor cost

**System Responses:** the list of countries with labor cost more than $10000 are displayed

**Post Condition:** System displays the list of countries for the condition

**Error:** No country available

**Query:**

SELECT C.Country, C.Code, C.Industry,D.`Avg\_Labour cost (USD/month)` FROM

((

SELECT A.Country, A.Code, B.Industry FROM

(

(SELECT \* FROM dmdd.country\_code)A

INNER JOIN

(SELECT \* from dmdd.dominant\_industry)B

on A.Country = B.Country

)

)C

inner join

(SELECT \* FROM dmdd.labour\_cost)D

on C.Country=D.Country

)

having D.`Avg\_Labour cost (USD/month)`<500;

1. **Use Case:** Identify the user who joined twitter in the last year with their highest follower count and corresponding tweet about India

**Description:** User views the user’s name who mentioned a particular country in this case it is India

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the user’s name who mentioned a particular country in this case it is India

**System Responses:** the list of user names who mentioned a particular country

**Post Condition:** System displays the list of user names for the condition

**Error:** No users available

**Query:**

select B.tweet\_user\_id,A.joining\_date, A.user\_name,max(A.follower\_count),B.tweet\_content from

(select \* from dmdd.twitter\_users

where follower\_count<10)A

inner join

(select tweet\_user\_id, tweet\_content from dmdd.tweets

group by 1)B

on A.user\_id=B.tweet\_user\_id

where B.tweet\_content like '%India';

1. **Use Case:** View the user who tweeted about #GDP and find the count of tweets having #GDP

**Description:** User views the user’s name with a particular following count in this case it is 2

**Actor:** User

**Precondition:** User must have access to data

**Steps:**

**Actor action:** User views the user’s name whose following count is more than 2

**System Responses:** the list of user names who have more than 2 following

**Post Condition**: System displays the list of user names for the condition

**Error:** No users available

**Query:**

select A.tweet\_user\_id,B.tag,count(tweet\_content) from(

(select \* from dmdd.tweets)A

inner join

(select tweet\_id, tag from dmdd.tags where tag like '%GDP%')B

on A.tweet\_id=B.tweet\_id);

1. **Use Case:**Search for a country codes with respect to the domain that he ventures in example Oil and get the country offering with lowest labor cost

**Description:** User makes a search for country with lowest labour cost of business he wants to establish

**Actors:**User

**Precondition:** User must have access to database

**Steps:**

**Actor action** – User tweets about his domain of business that he is interested in

**System Responses** – Response with respect to the domain of user choice is generated

**Post Condition:**The specific response for the search type is added into the database for table

**Alternate Path:**The specific search is not available

**Error:**Search Not Available

**SQL:**

SELECT f.Code, e.`Avg\_Labour cost (USD/month)` from DMDD.dominant\_industry as d Inner Join DMDD.labour\_cost as e on d.Country=e.Country Inner Join DMDD.Country\_Code as f on d.country=f.country Where d.Industry='Oil' group by f.Code order by e.`Avg\_Labour cost (USD/month)` Limit 1;

**Relation Algebra:**

Labor\_cost<-  σ Industry=”Oil”(dominant\_industry)

Country\_Code <-. Labor\_cost country=country(CountryCode)

Res <- π code,labor\_cost(Country\_Code)

1. **Use Case:** Generate the country codes with lowest labor cost and their respective GDP

**Description:** User makes a search for country with the lowest labor cost along with their GDP

**Actors:** User

**Steps:**

**Actor** action – User views the country with lowest labor cost

**System Responses** – the country with lowest labor cost is displayed

**Post Condition:** system displays specific country with the lowest labor cost

**SQL:**

SELECT f.Code,(e.`Avg\_Labour cost (USD/month)`),d.`GDP(Trillion $)` from DMDD.Economic\_Ranking as d Inner Join DMDD.labour\_cost as e

on d.Country=e.Country

Inner Join DMDD.Country\_Code as f

on d.country=f.country

where e.`Avg\_Labour cost (USD/month)`=(select min(`Avg\_Labour cost (USD/month)`) from DMDD.labour\_cost);

**Relation Algebra:**

Economic\_Ranking<-  σ Avg\_Labor\_cost=Min(Avg\_labor\_cost) (Labor\_Cost)

Country\_Code <-.Economic\_Ranking country=country (CountryCode)

Res <- π country(Country\_Code)

1. **Use Case:**View the country with top 10 gdp, their country codes and their respective industry

**Description:** User views the top 10 country as per GDP and their corresponding industry

**Actor:** User

**Actor action:** User views the country their ,GDP and corresponding industry

**System Responses:** the list of top 10 countries is displayed

**Post Condition:** system displays the list of countries for the condition

**SQL:**

SELECT f.Code,f.Country,e.Industry,d.`GDP(Trillion $)` from DMDD.Economic\_Ranking as d Inner Join DMDD.dominant\_industry as e

on d.Country=e.Country

Inner Join DMDD.Country\_Code as f

on d.country=f.country

order by d.`GDP(Trillion $)` desc Limit 10

**Relation Algebra:**

Country\_code<-  σ Economy\_Ranking=GDP (Country)

Dominant\_Industry <-.Country\_code country=country (Industry)

Res <- π code,country,industry (Dominant\_industry)

1. **Use Case:**View the user’s who joined in last month and who has the most likes and identify the tweet with most likes

**Description:** User views the user’s name with a particular follower count in this case 2

**Actor**: User

**Precondition:**User must have access to data

**Steps:**

**Actor action:** User views the user’s name whose following count is more than 2

made by a user

**Post Condition:** System displays the list of user names for the condition

**Error:** No users available

**SQL:**

SELECT tu.user\_id,tu.joining\_date,tw.Likes,tw.tweet\_content FROM DMDD.tags as ta

INNER JOIN DMDD.tweets as tw

ON ta.tweet\_id = tw.tweet\_id

INNER JOIN DMDD.twitter\_users as tu

ON tw.tweet\_user\_id=tu.user\_id

where month(tu.joining\_date)=month(now())-1

order by tw.Likes desc limit 1;

**Relation Algebra:**

Userid <-  σ likes=Tweets (Tweetid)

Joiningdate <-.Userid userid=userid Twitter\_users

Res <- π Joiningdate,likes (Userid)

1. **Use Case:**Search all the tweets in last 1 month which were tweeted about gdp and give their tags

**Description:** User views all the tweets created between two specific dates

**Actor**: User

**Precondition:** Users should have access to the data

**Actor action:**User views the tweets between two specific dates

**System Responses:**the list of tweets will be generated as per the condition

**Post Condition:**system displays the list of tweets for the condition

**Error:** no tweets available

**SQL:**

SELECT ta.tag,tw.tweet\_created\_at,tw.tweet\_id,tu.user\_id FROM DMDD.tags as ta

INNER JOIN DMDD.tweets as tw

ON ta.tweet\_id = tw.tweet\_id

INNER JOIN DMDD.twitter\_users as tu

ON tw.tweet\_user\_id=tu.user\_id

where month(tw.tweet\_created\_at)=month(now())-1;

**Relation Algebra:**

Userid <-  σ TweetContent=GDP (Tweets)

tweetCreateat <-.Userid userid=userid Tweet

Res <- π createdat,Userid (tweeCreateat)

1. **Use Case:** View the list of popular industries among all the countries with a GDP rank > 50 and retrieve the corresponding country code

**Description:** User views the popular industry(the industry that is repeated the max number of times) and retrieves the country code

**Actors:** User

**Precondition:** User must have access to the data

**Steps:**

**Actor** **action** – User views a country and its leading industry/sector with GDP>3billion

**System Responses** – The name of industry would be displayed

**Post Condition:** the industry name will be displayed and country code will be displayed

**Query:**

SELECT D.Code, C.Country, C.Ranking, C.Industry FROM

((

SELECT A.Country, A.Ranking, B.Industry FROM

(

(SELECT \* FROM dmdd.economic\_ranking)A

INNER JOIN

(SELECT \* from dmdd.dominant\_industry)B

on A.Country = B.Country

)

)C

inner join

(SELECT \* FROM dmdd.country\_code)D

on C.Country=D.Country

)

where Ranking>50;

**Relation Algebra:**

πD.Code, C.Country, C.Ranking, C.Industry (A.Country, A.Ranking, B.Industry ) ⋈ π dmdd.economic\_ranking (dmdd.economic\_ranking) π ϱ dmdd.economic\_ranking( A) ⋈ ϱ dmdd.dominant\_industry (B) ϱ A.Country = B.Country ( C) ⋈ ϱ dmdd.country\_code (D) π C.Country=D.Country σ Ranking>50

1. **Use Case:** View the country and country code with the lowest average labour costs where textile sector can be set up

**Description:** User views the lowest average labor cost country

**Actors:** User

**Precondition:** User must have access to the Labor costs data and the industry data of the countries

**Steps:**

**Actor action** – User views a country with the lowest labor costs with textile industry

**System Responses** – The name of the country with lowest labor cost will be displayed

**Post Condition**: the country with lowest labor cost will be displayed with textile industry

**QUERY:**

SELECT D.Code, C.Country, min(C.`Avg\_Labour cost (USD/month)`), C.Industry FROM

(SELECT A.Country, A.`Avg\_Labour cost (USD/month)`, B.Industry FROM

((SELECT \* FROM dmdd.labour\_cost)A

INNER JOIN

(SELECT \* from dmdd.dominant\_industry)B

on A.Country = B.Country

))C

inner join

(SELECT \* FROM dmdd.country\_code)D

on C.Country=D.Country

**Relation Algebra:**

πD.Code, C.Country, C.Ranking, C.Industry (A.Country, A.Ranking, B.Industry X dmdd.economic\_ranking ) ⋈ π dmdd.dominant\_industry σ (A.Country = B.Country) ⋈ π dmdd.country\_code ( C.Country=D.Country)

1. **Use Case:** View the GDP rankings of the top 5 countries with the highest labor costs and respective leading industries

**Description: User views the top 5 countries with the highest labor costs and their** corresponding GDP rankings and leading industries

**Actors:** User

**Precondition:** User must have access to the Labor costs data, GDP ranking data, leading industries

**Steps:**

**Actor action** – User views the top 5 countries with the highest labor costs and corresponding GDP rankings

**System Responses** – The list of countries with highest labor costs combined with the GDP rankings and leading industries is displayed

**Post Condition:** Top 5 countries with highest labor costs combined with the GDP rankings and leading industries is displayed

**QUERY:**

SELECT C.Country,D.ranking, C.Industry, (C.`Avg\_Labour cost (USD/month)`) FROM

((SELECT A.Country, A.`Avg\_Labour cost (USD/month)`, B.Industry FROM

((SELECT \* FROM dmdd.labour\_cost)A

INNER JOIN

(SELECT \* from dmdd.dominant\_industry)B

on A.Country = B.Country

))C

inner join

(SELECT \* FROM dmdd.economic\_ranking)D

on C.Country=D.Country

)

order by (C.`Avg\_Labour cost (USD/month)`) desc limit 5

**Relation Algebra:**

πC.Country,D.ranking, C.Industry (A.Country, A.`Avg\_Labour) ⋈ π B.Industry (dmdd.labour\_cost) π ϱ dmdd.economic\_ranking( A) ⋈ ϱ dmdd.dominant\_industry (B) ϱ A.Country = B.Country ( C) ⋈ ϱ dmdd.economic\_ranking (D) π C.Country=D.Country τ (C.`Avg\_Labour cost (USD/month)`) desc limit 5

1. **Use Case:** Twitter IDs of users with less than follower count of 10 and get the total tweet count of the user

**Description:** Find users with follower count less than 10 and identify the corresponding number of total tweets of the user

**Actors:** User

**Precondition:** User must have access to twitter to tweet

**Steps:**

**Actor action** – User updates the GDP values of the top 5 ranked countries

**System Responses** –Twitter IDs of users, follower count, total tweet count of the user

**Post Condition:**

**Query:**

select B.tweet\_user\_id, A.user\_name,A.follower\_count, B.Tot\_tweets from

(select \* from dmdd.twitter\_users

where follower\_count<10)A

inner join

(select tweet\_user\_id, count(tweet\_content) Tot\_tweets from dmdd.tweets

group by 1)B

on A.user\_id=B.tweet\_user\_id;

**Relation Algebra:**

π B.tweet\_user\_id, A.user\_name,A.follower\_count, B.Tot\_tweets(dmdd.twitter\_users

) σ follower\_count<10 ) ⋈

π tweet\_user\_id, count, Tot\_tweets (dmdd.tweets) ℑ 1 ⋈ (A.user\_id=B.tweet\_user\_id)

1. **Use Case:** Which user tweeted about the most popular tag and get all the user twitter account information

**Description:** FInd the twitter handle and username for the user updating about the respective sectors

**Actors:** User

**Precondition:** User must have access to twitter to tweet

**Steps:**

**Actor action –** User updates about the Textile, Elctronic and Agriculture sectors

**System Responses –** The countries with the Textile, Elctronic and Agriculture sectors as their leading sectors will be displayed

**Post Condition:** Countries with the Textile, Elctronic and Agriculture sectors as their leading sectors will be displayed

**QUERY:**

select \* from

(

select B.tweet\_id,A.tag,B.tweet\_user\_id,B.tweet\_created\_at, B.Likes from

(select \* from dmdd.tags)A

inner join

(select \* from dmdd.tweets

)B

on A.tweet\_id=B.tweet\_id

)C

inner join

(select \* from dmdd.twitter\_users)D

on C.tweet\_user\_id=D.user\_id

where C.tag=(select (L.tag) from(

select K.tag,max(K.tot\_tags) from

(select tag,(count(\*)) tot\_tags from dmdd.tags

group by tag)K)L);

**Relation Algebra:**

π B.tweet\_id,A.tag,B.tweet\_user\_id,B.tweet\_created\_at, B.Likes (dmdd.tags

) ⋈ π dmdd.tweets (A.tweet\_id=B.tweet\_id) ⋈ πdmdd.twitter\_users(C.tweet\_user\_id=D.user\_id) σ C.tag=(select (L.tag) (K.tag,max(K.tot\_tags) ℑ tag

**Basic SQL on the general questions to be asked:Queries you must answer about your physical model (In Relational algebra & SQL):**

1. **What user posted this tweet?**

**SQL Query:**

select user\_name, tweet\_content from

(select user\_id, user\_name from dmdd.twitter\_users)A

inner join

(select tweet\_user\_id, tweet\_content from dmdd.tweets)B

on A.user\_id=B.tweet\_user\_id

**Relation Algebra:**

π user\_name, tweet\_content (user\_id, user\_name from dmdd.twitter\_users)ϱ A ⋈ π tweet\_user\_id, tweet\_content (dmdd.tweets) )ϱ B π A.user\_id=B.tweet\_user\_id

1. When did the user post this tweet?

**SQL Query:**

select user\_name, tweet\_content, tweet\_created\_at from

(select user\_id, user\_name from dmdd.twitter\_users)A

inner join

(select tweet\_user\_id, tweet\_content, tweet\_created\_at from dmdd.tweets)B

on A.user\_id=B.tweet\_user\_id

**Relation Algebra:**

π user\_name, tweet\_content, tweet\_created (user\_id, user\_name , dmdd.twitter\_users)ϱ A ⋈ π tweet\_user\_id, tweet\_content, tweet\_created (dmdd.tweets) )ϱ B π A.user\_id=B.tweet\_user\_id

1. **What tweets have this user posted in the past 24 hours?**

**SQL Query:**

select user\_name, tweet\_content, tweet\_created\_at from

(select user\_id, user\_name from dmdd.twitter\_users)A

inner join

(select tweet\_user\_id, tweet\_content, tweet\_created\_at from dmdd.tweets)B

on A.user\_id=B.tweet\_user\_id

where tweet\_created\_at=tweet\_created\_at-1

**Relation Algebra:**

π user\_name, tweet\_content, tweet\_created (user\_id, user\_name ,dmdd.twitter\_users)ϱ A ⋈ π tweet\_user\_id, tweet\_content, tweet\_created (dmdd.tweets) )ϱ B π A.user\_id=B.tweet\_user\_id σ tweet\_created\_at=tweet\_created\_at-1

1. **How many tweets have this user posted in the past 24 hours?**

**SQL Query:**

select user\_name, count(tweet\_content), tweet\_created\_at from

(select user\_id, user\_name from dmdd.twitter\_users)A

inner join

(select tweet\_user\_id, tweet\_content, tweet\_created\_at from dmdd.tweets)B

on A.user\_id=B.tweet\_user\_id

group by user\_name

**Relation Algebra:**

π user\_name, count, tweet\_created user\_id, user\_name ,dmdd.twitter\_users)ϱ A ⋈ π tweet\_user\_id, tweet\_content, tweet\_created (dmdd.tweets) )ϱ B π A.user\_id=B.tweet\_user\_id ℑ user\_name

1. **When did this user join Twitter?**

**SQL Query:**

select distinct(user\_name),joining\_date from

(select user\_id, user\_name, joining\_date from dmdd.twitter\_users)A

inner join

(select tweet\_user\_id, tweet\_content, tweet\_created\_at from dmdd.tweets)B

on A.user\_id=B.tweet\_user\_id

**Relation Algebra:**

π distinct ,joining\_date (user\_id, user\_name, joining\_date, dmdd.twitter\_users)ϱ A ⋈ tweet\_user\_id, tweet\_content, tweet\_created (dmdd.tweets) )ϱ B π A.user\_id=B.tweet\_user\_id

1. **What keywords/ hashtags are popular?**

The hastag used the most number of times is the popular hashtag.

**SQL Query:**

select A.tag, max(A.tot\_tags) from(

select tag,(count(\*)) tot\_tags from dmdd.tags

group by tag)A;

1. **What tweets are popular?**

Tweets with the most number of likes are popular.

**SQL Query:**

select user\_name, tweet\_content, max(likes) from

(select user\_id, user\_name, joining\_date from dmdd.twitter\_users)A

inner join

(select tweet\_user\_id, tweet\_content, tweet\_created\_at, likes from dmdd.tweets)B

on A.user\_id=B.tweet\_user\_id;

MIT License

Copyright (c) 2022 Mansi Umakant Dandgaval, Ajay Sureka, Gunasekhar Vinugolu, Akhila Boppana

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

Material referred to:

<https://github.com/aiskunks/Jobs_Database>

<https://www.youtube.com/watch?v=kbKty5ZVKMY>

<https://www.youtube.com/watch?v=3KaffTIZ5II>

<https://www.youtube.com/watch?v=8PJGw123zeE>

<https://blog.devart.com/create-er-diagram-for-sql-server-database-using-ssms-and-sql-designer.html>