

**Entrega # 3 proyecto ingeniería de datos**

Alumno: Abraham Espinosa Galván

Profesor: Luz Stella Garcia Monsalve

Enlace del repositorio de Github: [eleibraaa/proyecto\_ingdatos](https://github.com/eleibraaa/proyecto_ingdatos)

**Resumen del proyecto**

El proyecto consiste en crear una base de datos relacionada con los elementos y entidades que se encuentran en nuestro sistema solar. Para ello usaremos tres bases de datos distintas que nos proporcionará información relevante acerca de los cuerpos celestes y objetos artificiales que se encuentran en nuestro sistema solar. Normalizaremos cada una de las bases de datos y se implementarán en PostgreSQL y finalmente se realizará una implementación con módulos de Python para representarlo en una página web.

**Sección de discusión sobre las consultas**

**Primera consulta:** Se realizo la visualización y el dash, sin embargo, se decidió cambiar la ecuación usada y se hallo el radio de los cuerpos celestes, ya que la falta de datos de la base datos daba una consulta muy pequeña. La principal ventaja de esta consulta es que permite hallar datos de los cuerpos celestes que no vienen incluido en la base de datos. Como principal desventaja es que se trata de una aproximación con un margen de error medio y no todos los cuerpos celestes se les pudo hallar el radio por motivos de falta de datos de la misma base de datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Usando

**Segunda consulta:** En la tabla de los NEO se realizaron 2 dash para comparar por tamaño los NEO que son potencialmente peligrosos para la tierra y para los que no. Para eso se tomo el promedio de los dos diámetros junto a su id y por medio de un grafico de barras se hizo la comparación. Esto permite ver si los NEO mas grandes son mas peligroso, sin embargo, solo se toma como variable el tamaño de estos y no la velocidad u otros factores.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Sitio web

Descripción generada automáticamente

**Tercera consulta:** Usando agregación avanzada y group by se hizo un ranking sobre la tabla contractor y contractor\_countries para identificar que empresas y países poseen la mayor cantidad de satélites. La ventaja es que la consulta es muy completa y se ve claramente la correlación entre las 2 consultas incluso, aunque, se podría incluir la tabla register para complementarla aún más.

SELECT

country AS Country,

COUNT(\*) AS Satellite\_Count

FROM (

SELECT

COSPAR\_number,

NORAD\_number,

country1 AS country

FROM

Contractor\_country

UNION ALL

SELECT

COSPAR\_number,

NORAD\_number,

 country2 AS country

FROM

Contractor\_country

UNION ALL

SELECT

COSPAR\_number,

NORAD\_number,

country3 AS country

FROM

Contractor\_country

UNION ALL

SELECT

COSPAR\_number,

NORAD\_number,

country4 AS country

FROM

Contractor\_country

UNION ALL

SELECT

COSPAR\_number,

NORAD\_number,

country5 AS country

FROM

Contractor\_country

) AS countries

JOIN

Satelite\_artificial sa ON countries.COSPAR\_number = sa.COSPAR\_number AND countries.NORAD\_number = sa.NORAD\_number

GROUP BY

country

ORDER BY

Satellite\_Count DESC;

**Cuarta consulta:** La cuarta consulta trata sobre la recursividad dentro de la tabla Cuerpo\_celeste, la ventaja de esta consulta es que en robusta y muestra resultados acertados a pesar de la complejidad de la consulta misma, with recursive OrbitHierarchy as (

select

 cc.id\_cl as planet\_id,

cc.orbits as orbits\_id

from Cuerpo\_celeste cc

where cc.orbits is not null -- Sin orbita

union all

-- Recursivo

select

oh.planet\_id, -- Propagar el ID del cuerpo inicial

cc.orbits as orbits\_id -- Buscar la órbita del siguiente nivel

from OrbitHierarchy oh

join Cuerpo\_celeste cc on oh.orbits\_id = cc.id\_cl

)

--Final

select

tdc\_planet.name as "Nombre del Planeta", -- Nombre del planeta

oh.planet\_id as "ID del Planeta", -- ID del planeta

tdc\_orbit.name as "Órbita del Cuerpo" -- Nombre del cuerpo que orbita

from OrbitHierarchy oh

left join Tipo\_de\_cuerpo tdc\_planet on oh.planet\_id = tdc\_planet.id\_d

left join Cuerpo\_celeste cc\_orbit on oh.orbits\_id = cc\_orbit.id\_cl

left join Tipo\_de\_cuerpo tdc\_orbit on cc\_orbit.id\_cl = tdc\_orbit.id\_d;

select

cc.id\_cl,

cc.orbits,

tdc.name as planet\_name

from Cuerpo\_celeste cc

left join Tipo\_de\_cuerpo tdc on cc.id\_cl = tdc.id\_d

left join Cuerpo\_celeste cc\_orbit on cc.orbits = cc\_orbit.id\_cl

left join Tipo\_de\_cuerpo tdc\_orbit on cc\_orbit.id\_cl = tdc\_orbit.id\_d;

PD: Las visualizaciones estan dentro del github

**Opiniones**

Respecto al resultado del proyecto, creo que se logró el resultado esperado. Diría que se podría mejorar la decoración de la página HTML, pero, por lo demás, el desarrollo de los módulos en Python, dash y consultas estuvo muy bien. Creo que fue un reto de buen nivel y de aprendizaje profundo acerca de cómo funciona una base de datos.