

**Entrega # 2 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.

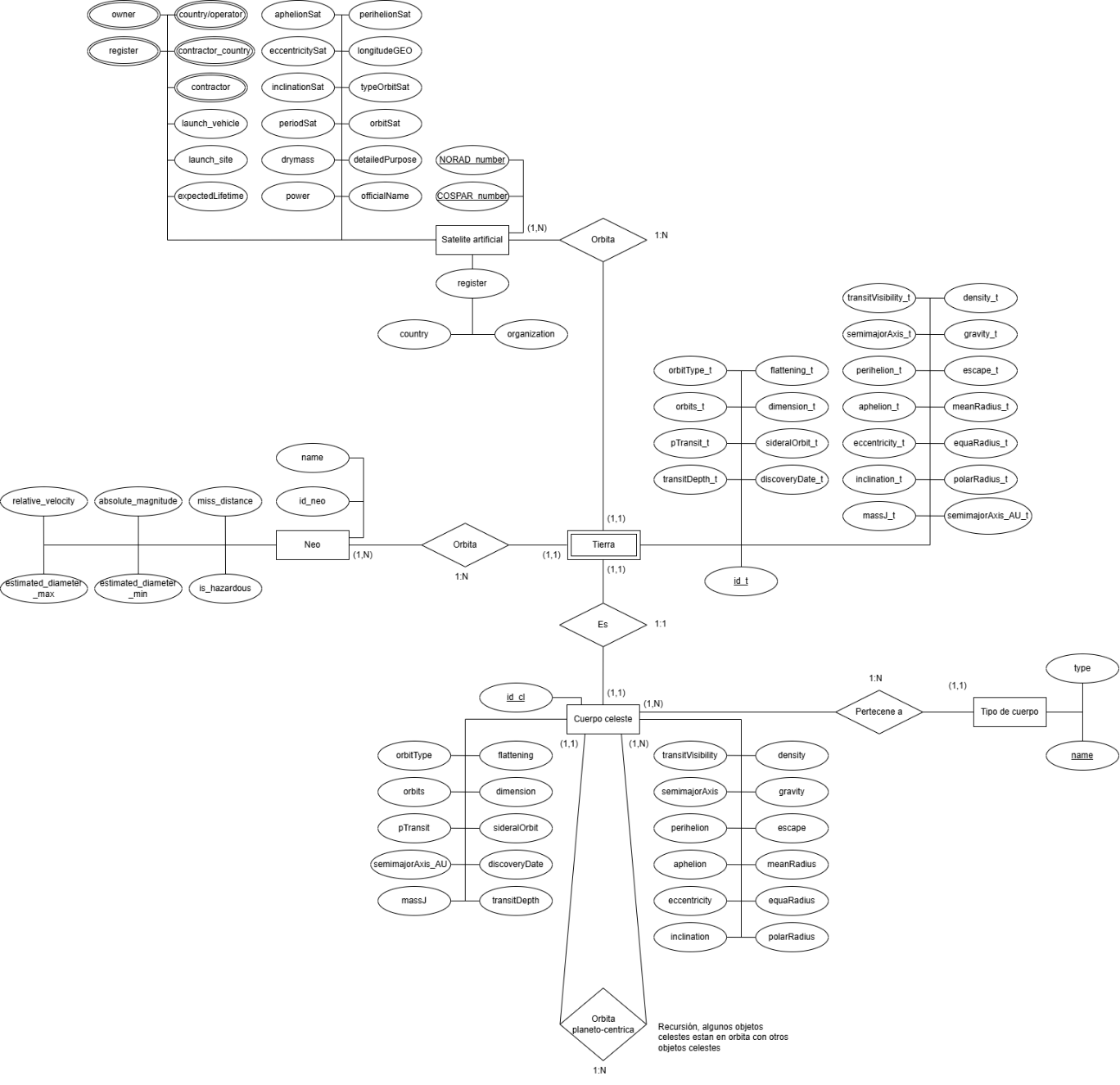
**Diagrama Relacional**

Diagrama

Descripción generada automáticamenteEl diagrama relacional cuenta con diez tablas en total, 5 de ellas son resultado de la atomización de las tuplas pertenecientes a Satelite\_artificial, las demás tablas son iguales a las presentadas en la primera entrega, sin embargo, algunos tipos de datos han sido cambiados para poder cargar masivamente los datos de los archivos *csv.*

**Modelo entidad relación**

A continuación, se mostrará el modelo entidad relación corregido de la primera entrega.



**Implementación de la base de datos a PostgreSQL**

Lo primero que se hizo fue analizar los archivos *csv* y eliminar las tuplas o columnas que contribuirán información corrupta o equivocada, este proceso se le realizó a todos los archivos *csv* *utf8* encontrados en el repositorio. Luego se crearon en una base de datos de SQL las tablas expuestas en el modelo relacional. Posteriormente, por medio de la sentencia a *copy* se cargan masivamente los datos de los archivos *csv* a las tablas de nuestra base de datos, a continuación. El código usado para la carga masiva también se encuentra en el repositorio junto a los *csv utf8. Este* código muestra cómo se fueron escribiendo las sentencias para poder cargar las tuplas a nuestra base de datos.

**Módulos de Python**

Los módulos de Python se realizaron para poder crear una conexión entre nuestra base de datos y Python. Se realizaron algunas consultas relacionadas con las tablas de la base de datos junto a la conexión de estas mismas al intérprete de Python. Para esto se usó la librería *psycopg2* que permite la conexión entre PostgreSQL y Python. El código de los módulos de Python se encuentra en el repositorio. El código está comentado para indicar la conexión de cada una de las tablas y el procedimiento que se siguió.

**Análisis de escenarios**

1. En la base de datos sobre el sistema solar se presentó en medio de la carga de datos que algunos datos eran demasiado grandes para ser exportados desde *csv utf8* a PostgreSQL, sin embargo, a través de triggers y funciones se pueden hallar usando fórmulas de la física clásica estos datos, tales como la masa o el volumen de un planeta.

(Ley de gravitación universal)

O para el volumen

(Segunda ley de Kepler)

Estas ecuaciones se pueden colocar como funciones y así poder obtener los datos adicionales del planeta usando los presentes en las tuplas, para así mostrar los resultados en un diagrama de barras u otros.

1. La tabla NEO está relacionada con la tabla Tierra por medio de id\_t. Una de las columnas de esta tabla es *isHazardous* que indica si el NEO es potencialmente peligroso para la tierra. Una consulta interesante puede ser la relación que tienen los atributos de los NEO y si en sí son peligrosos o no, ¿qué hace a un NEO peligroso? Su velocidad, tamaño, entre otros. Eso se buscará con la consulta, usando sentencias como *between* o mayor o menor, que se puede realizar una comparación entre los NEO peligrosos y los NEO que no por medio de sus atributos.
2. Satélites\_artificiales es una tabla con muchas tablas relacionadas con ella, por medio de *COSPAR\_number* y *NORAD\_number*, las tablas a las que hace referencia contienen tuplas acerca de los operadores y nacionalidades de los dueños de los satélites. Usando sentencias a agregación y agregación avanzada se puede hacer un ranking de que países o que empresas tienen o poseen la mayor cantidad de satélites en órbita.
3. En la tabla Cuerpo\_celeste está la columna *orbits. Esta* permite saber dependiendo del cuerpo celeste seleccionado a qué otro cuerpo celeste está orbitando. Este mismo cuerpo celeste al que orbita se encuentra en la misma tabla, formando así una relación recursiva. Entonces, a través de la recursividad y el uso de funciones se podría saber que cuerpos celestes orbitan a otros.