Big Data: Known Asteroids

Max Oakes

Last updated 3/11/22

What is in this dataset?

- Comes from https://www.kaggle.com/sakhawat18/asteroid-dataset
- Asteroids that have been observed, and many orbital statistics and physical properties
- Contains about 950,000 objects
- ~45 columns
- Single table
- Half of the columns did not have descriptions from the author

Obj internal ID	Primary SPK-ID	Full name/ designation	Primary designation
Eccentricity	Semi-major axis au Unit	perihelion distance au Unit	Inclination (degrees)
Absolute magnitude parameter	object diameter km Unit	Geometric albedo	1-sigma uncertainty in object diameter (km)
Time of perihelion passage TDB Unit	Earth Minimum Orbit Intersection Distance au Unit	Standard deviation of all metrics	IAU name
Prefix	Near-Earth Object	Potentially Hazardous Asteroid	Longitude of Ascending Node (Omega)
Arg. of Periapsis (omega)	Mean anomaly(?)	True anomaly (nu)	Solution ID
Epoch of osculation in modified Julian day form	Epoch, mean Julian date	Equinox of reference frame	Asteroid class abbreviation

Processing the dataset

- Source was in CSV format
- Wrote a Python script to write a CREATE TABLE statement for the data, and all INSERT INTO statements
- The script attempts to get the domain of the row, but manual review is needed
- Executing the python script on this dataset takes about 1 minute

```
    f = open(outputFile, "w")
    nameTypePairs = []
    for line in csv_reader:
        # print("Evaluating line %s" % (csv_reader.line_num))
        lineString = ""
        #create table
        if (csv_reader.line_num == 1):
            lineString = "CREATE TABLE %s (\n" % (sys.argv[2])

            #estimate row type (based on first data row), may need manual
adjustment
            nextLine = csv_reader.__next__()
            for i in range(0,len(line)):
                try:
                    t = int(nextLine[i])
                    # print("%s is an int" % (line[i]))
                    nameTypePairs.append((line[i],"int"))
                    continue
                except ValueError:
                    pass
                try:
                    t = float(nextLine[i])
                    # print("%s is a float" % (line[i]))
                    nameTypePairs.append((line[i],"float"))
                    continue
                except ValueError:
                    pass
                # print("%s must be a string" % (line[i]))
                nameTypePairs.append((line[i],"varchar(50)"))

            #write CREATE TABLE statement content
            for p in range(len(nameTypePairs)):
                if p != len(nameTypePairs)-1:
                    lineString = lineString + "\t" + " " +
str(nameTypePairs[p][0]) + " " + str(nameTypePairs[p][1]) + ",\n"
                else:
                    lineString = lineString + "\t" + " " +
str(nameTypePairs[p][0]) + " " + str(nameTypePairs[p][1]) + "\n);\n\n"
            print(lineString)
            f.write(lineString)
            print("Writing row data to file. Please wait...")

        #data rows
        else:
            lineString = "INSERT INTO %s VALUES (" % (sys.argv[2])
            #write INSERT INTO statements for each row
            for i in range(len(line)):
                item = line[i].strip()
                item = item.replace("'","''")
                #insert null if there is no item in the cell
                if (not item):
                    item = "NULL"
                else:
                    item = "'" + item + "'"
                #if it is a float value, make it in the correct notation
                try:
                    float(item)
                    item = format(item, '.12f')
                except:
                    pass

                #prepare the row for writing
                if i != len(line)-1:
                    lineString = lineString +  item + ","
                else:
                    lineString = lineString +  item + ");\n"

            #print(lineString)
            f.write(lineString)

    input_csv.close()
```

Processing the dataset

- Manually edited CREATE TABLE statement to update names to be more descriptive
- Added primary key
- Added enum for asteroid classification
- Used pgAdmin 4 as the query tool
- Ran the SQL script
    - Took about 5 minutes to insert all rows into the table

```
DROP TYPE IF EXISTS classification CASCADE;
CREATE TYPE classification AS ENUM
('AMO','APO','AST','ATE','CEN','HYA','IEO','IMB','MBA',
'MCA','OMB','PAA','TJN','TNO');

DROP TABLE IF EXISTS asteroids;
CREATE TABLE asteroids (
        id varchar(50),
        spkid int,
        full_name varchar(50),
        pdes varchar(50),
        fancy_name varchar(50),
        prefix varchar(50),
        neo boolean,
        pha boolean,
        absmag float,
        diameter float,
        albedo float,
        sigma_diameter float,
        orbit_id varchar(50),
        epoch float,
        epoch_mjd int,
        epoch_cal float,
        eccentricity float,
        semimajor_axis float,
        perihelion float,
        inclination float,
        asc_node_long float,
        arg_periapsis float,
        mean_anomaly float,
        ad float,
        true_anomaly float,
        time_peri_pass float,
        time_peri_calendar float,
        per float,
        per_y float,
        moid float,
        moid_ld float,
        sigma_eccentricity float,
        sigma_semimajor_axis float,
        sigma_perihelion float,
        sigma_inclination float,
        sigma_asc_node_long float,
        sigma_arg_periapsis float,
        sigma_mean_anomaly float,
        sigma_ad float,
        sigma_true_anomaly float,
        sigma_time_peri_pass float,
        sigma_per float,
        class classification,
        rms float,
    PRIMARY KEY (id)
);
```

Additional Modifications

- After creating and filling the table, I realized I can benefit from a second table that lists the classes of asteroids and their description
- I created an SQL script to create this table, and integrate that into the asteroid table

```
DROP TABLE IF EXISTS classes;
CREATE TABLE classes (
    id int NOT NULL,
    abbr varchar(5),
    name varchar(40),
    description varchar(200),
    PRIMARY KEY (id)
);

INSERT INTO classes VALUES ('1','AMO','Amor','Near-Eart
INSERT INTO classes VALUES ('2','APO','Apollo','Near-Ea
INSERT INTO classes VALUES ('3','AST','Asteroid','Aster
INSERT INTO classes VALUES ('4','ATE','Aten','Near-Eart
INSERT INTO classes VALUES ('5','CEN','Centaur','Object
INSERT INTO classes VALUES ('6','HYA','Hyperbolic Aster
INSERT INTO classes VALUES ('7','IEO','Interior Earth O
INSERT INTO classes VALUES ('8','IMB','Inner Main-belt
INSERT INTO classes VALUES ('9','MBA','Main-belt Astero
INSERT INTO classes VALUES ('10','MCA','Mars-crossing A
INSERT INTO classes VALUES ('11','OMB','Outer Main-belt
INSERT INTO classes VALUES ('12','PAA','Parabolic Aster
INSERT INTO classes VALUES ('13','TJN','Jupiter Trojan'
INSERT INTO classes VALUES ('14','TNO','Trans Neptunian

ALTER TABLE asteroids
ADD classID int;
ALTER TABLE asteroids ADD FOREIGN KEY(classID) REFERENC

UPDATE asteroids SET classID = 1 WHERE class = 'AMO';
UPDATE asteroids SET classID = 2 WHERE class = 'APO';
UPDATE asteroids SET classID = 3 WHERE class = 'AST';
UPDATE asteroids SET classID = 4 WHERE class = 'ATE';
UPDATE asteroids SET classID = 5 WHERE class = 'CEN';
UPDATE asteroids SET classID = 6 WHERE class = 'HYA';
UPDATE asteroids SET classID = 7 WHERE class = 'IEO';
UPDATE asteroids SET classID = 8 WHERE class = 'IMB';
UPDATE asteroids SET classID = 9 WHERE class = 'MBA';
UPDATE asteroids SET classID = 10 WHERE class = 'MCA';
UPDATE asteroids SET classID = 11 WHERE class = 'OMB';
UPDATE asteroids SET classID = 12 WHERE class = 'PAA';
UPDATE asteroids SET classID = 13 WHERE class = 'TJN';
UPDATE asteroids SET classID = 14 WHERE class = 'TNO';

ALTER TABLE asteroids DROP COLUMN class;
DROP TYPE classification;
```

20+ Questions

Question 1

How many asteroids have ‘formal’ names? (and list a few of them)

```
SELECT count(*)
FROM asteroids
WHERE fancy_name is not null;

SELECT fancy_name
FROM asteroids
WHERE fancy_name is not null
LIMIT 12;
```

count
22064

fancy_names
Ceres
Pallas
Juno
Vesta
Astraea
Hebe
Iris
Flora
Metis
Hygiea
Parthenope
Victoria

Basically, if there is a ‘fancy name’, the asteroid is named after something. Other asteroids have the name in a form like “4835 (1989 BQ)”

Question 2

How many asteroids do not have a recorded size? What percentage of the asteroids in the database do not have a size recorded?

```
SELECT count(*)
FROM asteroids
WHERE diameter is null;
```

count
8822315

```
SELECT (count(*)/(
    SELECT count(*) FROM
    asteroids)::FLOAT)*100 as
    not_recorded
FROM asteroids
WHERE diameter is null;
```

not_recorded
85.7897141855603

So, almost 86% of the asteroids in the database do not have a size recorded. Likely, we know what their orbit is, but we do not have a good estimate of the size.

Question 3

Of the asteroids that we know the sizes of, what are the 20 smallest ones and their sizes in meters?

```
SELECT full_name, diameter*1000 as
    diameter_m
FROM asteroids
ORDER BY diameter
LIMIT 20;
```

In the table, the units for diameter are km, so the query result needs to be adjusted a bit.

full_name	diameter_m
(2012 XB112)	2.5
(2010 FD6)	8
(2010 GH7)	8
(2010 KV7)	13
(2010 FR9)	15
(2010 FT)	18
(2010 TN4)	18
(2010 DL)	19
(2010 FS)	23
(2010 FW9)	24
(2010 YD)	26
(2002 JR100)	28
(2010 FX9)	30
(1998 KY26)	30
(2010 HA)	32
(2010 JJ3)	32
(2010 CO44)	34
(2010 JO71)	37
(2010 QG2)	38
(2010 JW39)	39

Question 4

What are the known top 20 largest asteroids and what is their diameter in miles?

```
SELECT full_name,
    round(CAST(diameter*0.621371 as
    numeric), 4) as diameter_mi
FROM asteroids
where diameter is not null
ORDER BY diameter DESC
LIMIT 20;
```

In the table, the units for diameter are km, so the query result needs to be adjusted a bit.

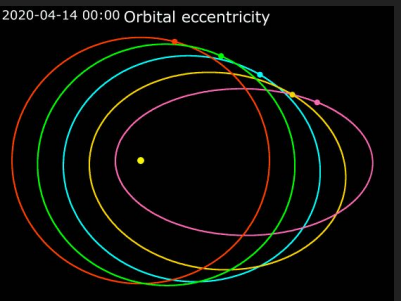
In order to truncate the many digits in a double precision, I had to cast to a numeric and truncate the digits using a function.

full_name	diameter_mi
1 Ceres	583.7159
20000 Varuna (2000 WR106)	559.2339
2 Pallas	338.6472
4 Vesta	326.4683
10 Hygiea	252.9726
15789 (1993 SC)	203.8097
704 Interamnia (1910 KU)	190.334
52 Europa	188.8458
10199 Chariklo (1997 CU26)	187.654
511 Davida (1903 LU)	167.9734
31 Euphrosyne	165.9558
451 Patientia (1899 EY)	157.7661
87 Sylvia	157.2386
3 Juno	153.2276
65 Cybele	147.4265
88 Thisbe	144.1581
15 Eunomia	143.9648
95626 (2002 GZ32)	143.226
16 Psyche	140.4298
624 Hektor (1907 XM)	139.8085

Question 5

Of the largest 1000 asteroids, which 15 have the smallest eccentricity?

```
SELECT full_name, diameter,
eccentricity
FROM (
    SELECT *
    FROM asteroids
    WHERE diameter is not null
    ORDER BY diameter DESC
    LIMIT 1000
) as largest
ORDER BY eccentricity
LIMIT 15;
```



Orbital eccentricity determines the amount by which its orbit around another body deviates from a perfect circle.

e = 0 is a circular orbit
0 < e < 1 is an elliptic orbit
e =1 is a parabolic escape orbit
e > 1 is a hyperbola.

full_name	diameter (in km)	eccentricity
1262 Sniadeckia (1933 FE)	71.011	0.00459534174
4754 Panthoos (5010 T-3)	53.025	0.007840463
508 Princetonia (1903 LQ)	117.241	0.008150032768
1308 Halleria (1931 EB)	46.951	0.01123433027
5130 Ilioneus (1989 SC7)	60.711	0.01126255871
208 Lacrimosa	40.056	0.01191309814
1838 Ursa (1971 UC)	40.054	0.01470903061
2207 Antenor (1977 QH1)	97.658	0.0148056264
15502 (1999 NV27)	53.1	0.01548989195
196 Philomela	144.626	0.01572406943
702 Alauda (1910 KQ)	190.98	0.01710190003
2223 Sarpedon (1977 TL3)	77.48	0.01772377249
4867 Polites (1989 SZ)	57.251	0.01871111508
1647 Menelaus (1957 MK)	42.716	0.02115375445
528 Rezia (1904 NS)	91.966	0.02117100109

Question 6

Of the 1000 asteroids with the highest eccentricity, what are the 20 that have the smallest perihelion in AU?

```
SELECT full_name, eccentricity,
perihelion
FROM (
    SELECT *
    FROM asteroids
    ORDER BY eccentricity DESC
    LIMIT 1000
) as widest
ORDER BY perihelion
LIMIT 20;
```

The perihelion is defined as the distance in an orbit where an object is closest to the sun.

For reference, 0.07 AU is about 10,500,000 km.

full_name	eccentricity	perihelion (in AU)
(2005 HC4)	0.9613212183	0.07051073204
(2020 BU13)	0.970394722	0.07312539742
(2017 TC1)	0.9695289871	0.07587192754
(2017 MM7)	0.9615701466	0.07924409951
(2008 FF5)	0.9651477164	0.0793792941
(2015 EV)	0.9602607237	0.08074429596
394130 (2006 HY51)	0.9683957212	0.08181996324
(2016 GU2)	0.9574974744	0.0873170149
(2019 JZ6)	0.9633213451	0.09079732509
(2019 AM13)	0.9297880881	0.09102190758
137924 (2000 BD19)	0.8950019842	0.09202463497
374158 (2004 UL)	0.926620592	0.09292416119
394392 (2007 EP88)	0.8858639314	0.09556534232
(2011 KE)	0.9545352937	0.1003088256
465402 (2008 HW1)	0.9599976925	0.1034743081
(2015 HG)	0.9500904688	0.1047956752
(2012 US68)	0.9579045737	0.105371839
(2011 XA3)	0.9259661436	0.10858392
(2018 GG5)	0.944725539	0.1098178443
399457 (2002 PD43)	0.9559445715	0.1104958793

Question 7

Of the asteroids flagged as potentially hazardous, what are the 10 that have the lowest eccentricity?

```
SELECT full_name, eccentricity
FROM asteroids
WHERE pha is true
ORDER BY eccentricity
LIMIT 10;
```

full_name	eccentricity
(2018 EB)	0.01217587595
(2005 TF49)	0.02545288003
(2011 DV)	0.04986977888
(2004 LB)	0.05288329937
(2014 WT202)	0.06301564512
474163 (1999 SO5)	0.06519877357
(2008 EE5)	0.07158705769
365071 (2009 AV)	0.07395467766
419624 (2010 SO16)	0.07542896203
385186 (1994 AW1)	0.07576828008

A potentially hazardous an asteroid (PHA) is an asteroid whose orbit comes nearer than 0.05AU (about 7.5 million km) to the Earth and whose brightness implies a size of the order of about 100m across or more.

"`WHERE pha`" could have been used instead, but the verbose query above is making it clear that PHA is a boolean value.

Question 8

Of the named asteroids that are not potentially hazardous, which one's names have the suffix '-eus'?

```
SELECT full_name, fancy_name
FROM asteroids
WHERE pha is not true and
fancy_name like '%eus';
```

Turns out there are only 36 of them.

"`WHERE not pha`" could have been used instead, but the verbose query above is making it clear that PHA is a boolean value.

full_name	fancy_name
2174 Asmodeus (1975 TA)	Asmodeus
2213 Meeus (1935 SO1)	Meeus
2759 Idomeneus (1980 GC)	Idomeneus
3793 Leonteus (1985 TE3)	Leonteus
4001 Ptolemaeus (1949 PV)	Ptolemaeus
4068 Menestheus (1973 SW)	Menestheus
4197 Morpheus (1982 TA)	Morpheus
5130 Ilioneus (1989 SC7)	Ilioneus
5259 Epeigeus (1989 BB1)	Epeigeus
5731 Zeus (1988 VP4)	Zeus
7152 Euneus (1973 SH1)	Euneus
7412 Linnaeus (1990 SL9)	Linnaeus
8125 Tyndareus (5493 T-2)	Tyndareus
8600 Arundinaceus (3060 T-2)	Arundinaceus
8752 Flammeus (2604 P-L)	Flammeus
8757 Cyaneus (6600 P-L)	Cyaneus
8968 Europaeus (1212 T-2)	Europaeus
9907 Oileus (6541 P-L)	Oileus
11311 Peleus (1993 XN2)	Peleus
12607 Alcaeus (2058 P-L)	Alcaeus
12916 Eteoneus (1998 TL15)	Eteoneus
14791 Atreus (1973 SU)	Atreus
20952 Tydeus (5151 T-2)	Tydeus
24587 Kapaneus (4613 T-2)	Kapaneus
24603 Mekistheus (1973 SQ)	Mekistheus
30704 Phegeus (3250 T-3)	Phegeus
32532 Thereus (2001 PT13)	Thereus
39463 Phyleus (1973 SZ)	Phyleus
58096 Oineus (1973 SC2)	Oineus
73637 Guneus (1973 SX1)	Guneus
1143 Odysseus (1930 BH)	Odysseus
136557 Neleus (5214 T-2)	Neleus
173086 Nireus (2007 RS8)	Nireus
188847 Rhipeus (2006 FT9)	Rhipeus
1809 Prometheus (2522 P-L)	Prometheus
1810 Epimetheus (4196 P-L)	Epimetheus

Question 9

How many near-earth objects are there that are not potentially hazardous?

```
SELECT count(*)
FROM asteroids
WHERE not pha and neo;
```

count
20828

```
SELECT count(*)
FROM asteroids
WHERE pha and not neo;
```

count
0

Bonus query! A potentially hazardous asteroid also needs to be considered an NEO. You cannot have a PHA that is not a NEO.

A near-Earth object is an asteroid or comet which passes close to the Earth's orbit. In technical terms, a NEO is considered to have a trajectory which brings it within 1.3 astronomical units of the Sun and hence within 0.3 astronomical units, or approximately 45 million kilometres, of the Earth's orbit.

Question 10

How many asteroids are named only after a year and an alphanumeric designation?

```
SELECT count(*)
FROM asteroids
WHERE full_name ~
    '^\([0-9]{4}\s[A-Z0-9\-]+\)';
```

count
413371

This query is mainly an exercise in regular expressions. This regular expression accepts a string like ‘(2003 TY29)’

Question 11

Of the 100 smallest asteroids, which 20 have the most uncertainty about their size (standard deviation)?

```
SELECT full_name, diameter,
sigma_diameter
FROM (
    SELECT *
    FROM asteroids
    ORDER BY diameter
    LIMIT 100
) as smallest
WHERE sigma_diameter is not null
ORDER BY sigma_diameter DESC
LIMIT 20;
```

full_name	diameter	sigma_diameter
(2014 VP35)	0.122	0.051
(2010 VT11)	0.152	0.044
(2010 LJ68)	0.193	0.037
(2014 RH12)	0.088	0.036
(2010 JG)	0.192	0.03
(2010 LJ61)	0.192	0.03
(2009 WA)	0.164	0.03
264357 (2000 AZ93)	0.113	0.029
(2010 HZ104)	0.14	0.025
(2010 GA7)	0.151	0.024
469445 (2002 LT24)	0.143	0.024
(2010 LL68)	0.153	0.024
(2011 AV55)	0.063	0.024
(2010 CF55)	0.176	0.022
(2010 GB6)	0.134	0.021
411165 (2010 DF1)	0.159	0.02
(2010 KA8)	0.183	0.019
(2010 LK61)	0.191	0.019
475016 (2005 UO)	0.164	0.019
(2010 WB)	0.057	0.018

Question 12

Of the different classes of asteroids, what is their count per class, average size, eccentricity, perihelion and albedo?

```
SELECT class, count(*),
avg(diameter) as avg_dia,
    avg(eccentricity) as avg_e,
    avg(perihelion) as avg_q,
    avg(albedo) as avg_albedo
FROM asteroids
GROUP BY class;
```

class	count	avg_dia	avg_e	avg_q	avg_albedo
AMO	8457	1.752	0.4044	1.1294	0.1732
APO	12687	0.9556	0.4869	0.8129	0.1736
AST	76	13.0441	0.4418	2.825	0.0632
ATE	1729	0.6157	0.3225	0.6075	0.2308
CEN	506	52.7312	0.4401	8.9468	0.0768
HYA	4	[null]	1.2645	5.0488	[null]
IEO	22	[null]	0.3443	0.4545	[null]
IMB	20360	2.2999	0.0757	1.7687	0.428
MBA	855954	5.0964	0.1479	2.2882	0.1335
MCA	18685	3.3286	0.2981	1.5531	0.1888
OMB	28355	8.7813	0.1428	2.889	0.0668
TJN	8221	20.7805	0.0738	4.8162	0.0739
TNO	3468	155.48	0.2212	35.9493	0.0617

For reference:

AMO	Amor
APO	Apollo
AST	Asteroid
ATE	Aten
CEN	Centaur
HYA	Hyperbolic Asteroid
IEO	Interior Earth Object

IMB	Inner Main-belt Asteroid
MBA	Main-belt Asteroid
MCA	Mars-crossing Asteroid
OMB	Outer Main-belt Asteroid
TJN	Jupiter Trojan
TNO	Trans Neptunian Object

Question 13

Of the asteroids that have an eccentricity of less than 0.5 and a perihelion of less than 1 AU, (and that have a recorded albedo) what is the average albedo per classification?

```
SELECT c.name, avg(albedo)
FROM asteroids as a join classes as
c on a.classid=c.id
WHERE eccentricity < 0.5
    and perihelion < 1.0
    and albedo is not null
GROUP BY c.name;
```

class	avg (albedo)
Apollo	0.18666480446927372
Aten	0.2327142857142857

Albedo refers to an object's measure of reflectivity, or intrinsic brightness. A white, perfectly reflecting surface has an albedo of 1.0; a black, perfectly absorbing surface has an albedo of 0.0.

Apollo Class: Asteroids which cross Earth's orbit with a period greater than 1 year.
Aten Class: Asteroids which cross Earth's orbit with a period less than 1 year.

Question 14

Of the asteroids that have an eccentricity of less than 0.5 (not really elongated) and a perihelion of greater than 1 AU, (and that have a recorded albedo) what is the average albedo per classification, and count per classification?

```
SELECT c.name, avg(albedo),
count(*)
FROM asteroids as a join classes as
c on a.classid=c.id
WHERE eccentricity < 0.5
    and perihelion > 1.0
    and albedo is not null
GROUP BY c.name;
```

class	Avg (albedo)	count
Amor	0.2061948052	154
Apollo	0.1558125	16
Asteroid	0.06233333333	6
Centaur	0.0785	28
Inner Main-belt Asteroid	0.4280305206	557
Jupiter Trojan	0.07392786973	1873
Main-belt Asteroid	0.133521741	124001
Mars-crossing Asteroid	0.194153869	336
Outer Main-belt Asteroid	0.06688943615	7431
Trans Neptunian Object	0.046	2

Question 15

What is the ratio in average asteroid size to those in the inner-main belt to the outer belt? (InnerBeltSize/OuterBeltSize)

```
SELECT (
     SELECT avg(diameter)
     FROM asteroids as a join classes as c
on a.classid=c.id
     WHERE c.abbr='IMB'
) / (
     SELECT avg(diameter)
     FROM asteroids as a join classes as c
on a.classid=c.id
     WHERE c.abbr='OMB'
) as ratio;
```

ratio
0.26191174314939014

We didn’t even need a FROM clause in the top level of the query.

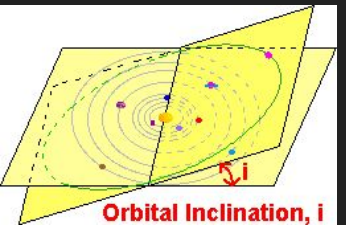
It looks like asteroids on the main belt closer to the sun are on average a fourth of the size of the asteroids on the outside of the main belt.

Question 16

How many trans-Neptunian objects are orbiting in retrograde (inclination greater than 90°)? And what is the average inclination of those objects?

```
SELECT count(*), avg(inclination)
FROM asteroids as a
     join classes as c on a.classid=c.id
WHERE c.abbr='TNO' and inclination > 90;
```

count	avg
51	129.26051965603924



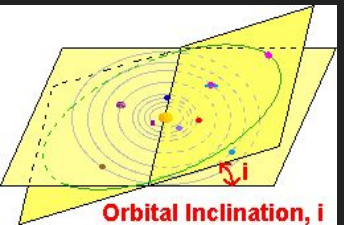
Inclination is the angle at which an object is orbiting a body.
When *i* = 0, it is an equatorial orbit,
When *i* = 90, it is a polar orbit,
When *i* > 90, it is a retrograde orbit.

Question 17

How many asteroids orbiting the sun are within 0.05 degrees of equatorial prograde? (i = 0 +/- 0.05^{o}), and what is the average eccentricity?

```
SELECT count(*), avg(eccentricity)
FROM asteroids
WHERE inclination < 0.05;
```

count	avg
56	0.19311544752753154



Inclination is the angle at which an object is orbiting a body.
When i = 0, it is an equatorial orbit,
When i = 90, it is a polar orbit,
When i > 90, it is a retrograde orbit.

Question 18

What are the known asteroids that have an inclination of less than 0.1º and an eccentricity of less than 0.1, (very circular orbit, and very ‘flat’ orbit) and what is their class?

```
SELECT c.name, count(*)
FROM asteroids as a
     join classes as c on a.classid=c.id
WHERE inclination < 0.1
and eccentricity < 0.1
GROUP BY c.name;
```

count	count
Main-belt Asteroid	23
Outer Main-belt Asteroid	2
Trans Neptunian Object	1

Question 19

What is the minimum, average and maximum inclination per class?

```
SELECT c.abbr, count(*),
     min(inclination), avg(inclination),
     max(inclination)
FROM asteroids as a
join classes as c on a.classid=c.id
GROUP BY c.abbr;
```

class	count	min	avg	max
AMO	8457	0.1320273752	13.55506268	159.0274588
APO	12687	0.02234662411	11.8152722	165.5410004
AST	76	1.653165083	21.79422131	163.1271026
ATE	1729	0.01351816639	12.50121499	65.83072462
CEN	506	0.9306363758	32.01648074	175.0829007
HYA	4	8.643584006	89.89551949	138.3809732
IEO	22	2.017520278	22.14155075	49.66151449
IMB	20360	0.7206441103	21.2906283	58.7443513
MBA	855954	0.007744219815	8.388177386	92.04431335
MCA	18685	0.09643805468	14.99666309	73.37315049
OMB	28355	0.03812908844	11.28143321	84.37279527
TJN	8221	0.1086573645	13.4208408	57.9108851
TNO	3468	0.03950137158	14.61498157	172.1361464

Question 20

How many objects have an inclination of 90 +/- 5 degrees? What is the average eccentricity?

```
SELECT count(*), avg(eccentricity)
FROM asteroids
WHERE inclination > 85 and
inclination < 95;
```

class	avg
21	0.7965265416017784

Asteroids that are close to a polar orbit appear to tend to have elongated orbits.

Question 21

What are all of the asteroids that are classified as ‘IEO’, but are not potentially hazardous?

```
SELECT *
FROM (
    SELECT full_name
    FROM asteroids as a join
classes as c on a.classid=c.id
    WHERE c.abbr='IEO'
except
    SELECT full_name
    FROM asteroids
    WHERE PHA
) as not_pha;
```

IEO = Interior Earth Object
An asteroid orbit contained entirely within the orbit of the Earth (Q < 0.983 AU).

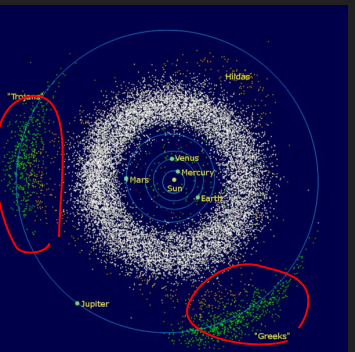
full_name
(2020 HA10)
(2006 WE4)
418265 (2008 EA32)
(2013 JX28)
164294 (2004 XZ130)
(2017 YH)
(2020 AV2)
(2015 ME131)
(2010 XB11)
(2018 JB3)
(2013 TQ5)
(2019 AQ3)
163693 Atira (2003 CP20)
(2019 LF6)
(1998 DK36)
413563 (2005 TG45)

Question 22

Of the Jupiter trojan asteroids, how many pairs of asteroids share the same diameter (within one meter)?

```
SELECT count(*)
FROM (
     asteroids as a1 join classes as c1
     on a1.classid=c1.id)
    join (asteroids as a2 join classes as c2
     on a2.classid=c2.id)
    on round(a1.diameter*1000)::
     int=round(a2.diameter*1000)::int
WHERE c1.abbr='TJN' and c2.abbr = 'TJN'
     and a1.diameter is not null
     and a2.diameter is not null
     and a1.id < a2.id;
```

count
104



When a1.id < a2.id, this will count each pair only once.
If it was '<>', it would count (a1, a2) and (a2,a1).

The Jupiter trojans are a large group of asteroids that share the planet Jupiter's orbit around the Sun.

There are 8221 Jupiter trojans in this table, for reference.

Question 23

In groups of asteroids that are binned by diameter in km, what are all of the bins have more than 100 asteroids?

```
SELECT count(*), ceil(diameter) as
     diameter_km_approx
FROM asteroids
GROUP BY ceil(diameter)
HAVING count(*) > 100
ORDER BY ceil(diameter) DESC;
```

count	diameter_ km_approx
822315	[null]
116	25
128	24
159	23
175	22
188	21
254	20
274	19
341	18
422	17
538	16
615	15
793	14
1049	13
1371	12
1773	11
2440	10
3566	9
5784	8
9278	7
14646	6
21599	5
28240	4
27603	3
12332	2
616	1

It appears that it is most common that an asteroid does not have a recorded diameter. Other than that, the 3-4km bin as the most amount of asteroids in it.

Question 24

What are the pairs (if any) of objects that share the same path, and what is the ‘distance’ between them (mean and true anomaly)?

```
SELECT a1.full_name, a2.full_name,
       abs(a1.true_anomaly-a2.true_anomaly) as
true_anomaly_diff,
       abs(a1.mean_anomaly-a2.mean_anomaly) as
mean_anomaly_diff
FROM (asteroids as a1 join asteroids a2
       ON (
              round(a1.semimajor_axis::numeric, 3)=
              round(a2.semimajor_axis::numeric, 3))
              and (round(a1.eccentricity::numeric, 3)=
              round(a2.eccentricity::numeric, 3))
              and (round(a1.inclination::numeric, 3)=
              round(a2.inclination::numeric, 3))
              and (round(a1.arg_periapsis::numeric, 3)=
              round(a2.arg_periapsis::numeric, 3))
              and (round(a1.asc_node_long::numeric, 3)=
              round(a2.asc_node_long::numeric, 3)))
WHERE a1.id < a2.id;
```

full_name	full_name	true_anomaly_diff	mean_anomaly_diff
(2013 EB88)	(2015 VU146)	1.16E-08	0.0003521971151
(2004 HC71)	(2011 GD94)	1.55E-08	5.15E-06
306381 (1993 RR2)	(2019 HC5)	4.13E-06	0.001162222932
534988 (2014 WF469)	(2006 BU292)	3.90E-08	0.000232868972

To determine an orbit’s path:

- a - SemiMajor Axis
- e - Eccentricity
- i - Inclination
- ω - Argument of the Periapsis
- Ω - Longitude of the Ascending Node

The result of this query show that these pairs have an extremely small distance between them. Perhaps these asteroids are closely orbiting each other, or these objects were measured and ‘found’ twice.