Department of Computer Science and Engineering

University of Dhaka

Project Report

CSE2211 - Database Management Systems-I

2nd Year 2nd Semester - 2018

Project Title

Astronomical Object Database

Submitted By

Name: Pranjal Kumar Nandi

Registration No: 2016-014-402

Roll No: JN-011

Introduction

Now a days, people of all ages are interested in gathering out-knowledge. Many young students are interested in Astronomical study as well as aged people. There are also some researcher who aren't contacted with any big research center. These people find difficulties to study about the space and celestial entity. Most of the time, it can be seen that many enthusiastic people left the study about them for only not getting the data in a specific place. They have to search a lot for getting any specific data. That's take a lot of time for them to study about them.

That's why it would be very much helpful if all the important data about the astronomical object can be stored in a specific place. This database project is made from that inspiration. It can save a lot of time and extra effort of the people interested in astronomical study.

The idea was to make a database which will contain the information of the astronomical entity such as star,planet and other type of things which are frequently asked. It is a smaller and simpler version. One can not get all types of data from here. But in spite of this, the database is seemed to be a good one for the beginner in the astronomical field.

Descriptions:

Project Idea:

My project idea was to build a database about astronomical object. It is more of like an encyclopedia type database where the information about the object like a star,a planet, a satellite or a galaxy, or even a constellation will be recorded so that one can find information from here. That's why the idea is named after "Astronomical Object Database".

Idea Details:

Firstly, the database has many entity. Planet, satellite, star, nebula all these objects are regarded as different entity. Because, every kind of them is different from others. So, related data and attributes will also be different.

First, the entity planet has some attributes. Such as, Earth is a planet. So earth has its mass, its radius, volume, temperature, co-ordination, age, orbital period, orbital speed, rotation period, rotation speed, satellite, galaxy, eccentricity etc. As a planet can have more then one satellite, there will be needed an another table. So satellite has been placed in that table with their value along with the planet they are orbiting.

Second, for the entity star, it has it's own name, mass, radius, temperature, surface pressure, volume, age, apparent magnitude, absolute magnitude, distance from earth, luminosity, rotation speed, galactic period etc and the galaxy it is in.

Now, for a galaxy there are several information including it's name, type, diameter, mass, angular momentum, red shift, est. number of stars, right ascension, declination, apparent magnitude, absolute magnitude, distance from solar system.

The next two objects are nebula and constellation. For the entity nebula, it's name, radius, distance, apparent dimension, declination, right ascension, galaxy name, constellation region have been implemented.

For the constellation, it's name, genitive, symbolism, right ascension, declination, area quadrant, main stars, number of stars with planet, number of flamsted stars, name of brightest star, messier objects and some other

relevant information have been added.

Main Features:

There are several features of the database project. Some are below here.

- This database will give us correct information about the astronomical object.
- One can query any single or more information from here for different purpose.
- Comparison between any two objects will be much more easier as the database is designed like this way.
- The attributes discussed in the idea details section have been implemented in the database. So, one can find all these data from the database by a single query.

Objectives:

The main objective of this database is to make the astronomical study easier for the mass people by providing information to them easily in a specific and single source of data which also can be called as a database. Finding a single data of a single entity such as a planet would be very difficult or time consuming when one has many short amount of time specially before the time before exam. The time one will waste by searching this information from a valid source, one can use it somewhere else to make it useful. So, this database can inspire people who have a short time for extra-curricular studies to be interested in astronomical side and also help the student by saving their time who will have a huge amount of study in the night before exam.

Motivation:

The main motivation behind this idea to create the astronomical study interesting to the mass people. As I was a child, space and related topics was always a huge interesting topics to me. But I faced the difficulties for getting the information. People can't buy a whole book just for some piece of information or neither can go to the library every time they need to find a answer of a question. So, how about a database where all the key information of a entity will be reserved. People will find it easy for getting the information. Even if one is in a travel or outside from home, one can easily search a data from it without any kind of extra pressure. This was the main reason of the idea.

Constraints:

- Right ascension value for every entity will be hour-minute-second format and have to be a string.
- Declination will be in degree-minute-second format and will be a string.
- Co-ordination will be in degree.
- Temperature will be in kelvin scale.
- Surface gravity will be in m/s^2 format. Surface area will be in km^2.
- Object's name will never be null and will always be in string.
- Eccentricity, age, number of any objects will be number.
- Distance for the star and galaxy will be in light-year and must be a number.
- Rotation time, orbital time, and all other kinds of time will be in years and must be a number. Rotation in own axis for the planet will be in days.
- Value of red-shift will be in km/s and must be a number.
- Mass for planet and star will be in kg but for galaxy it will be solar mass and both will be a number.

Design Diagrams:

Now, in this section schema and erd of the database project will be given.

Entity Relationship Diagram:

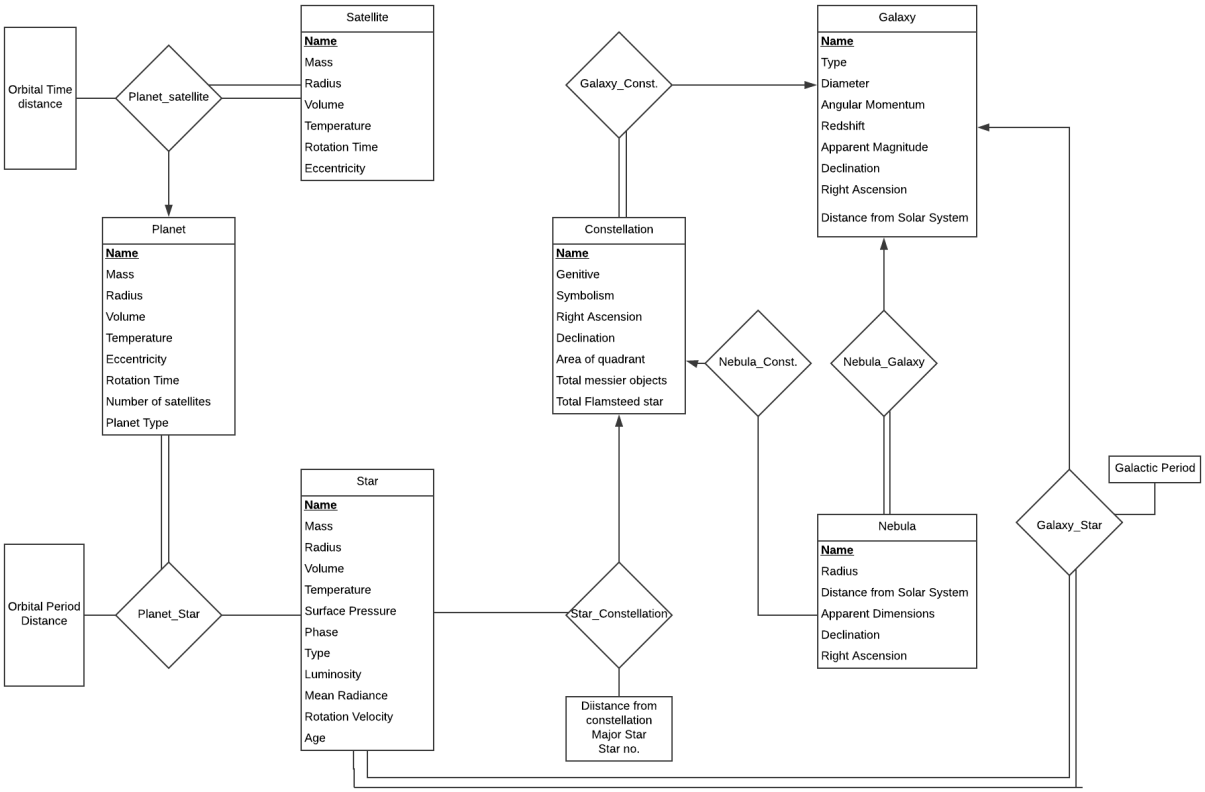

Fig: ERD of the database Project.

Schema Diagram:

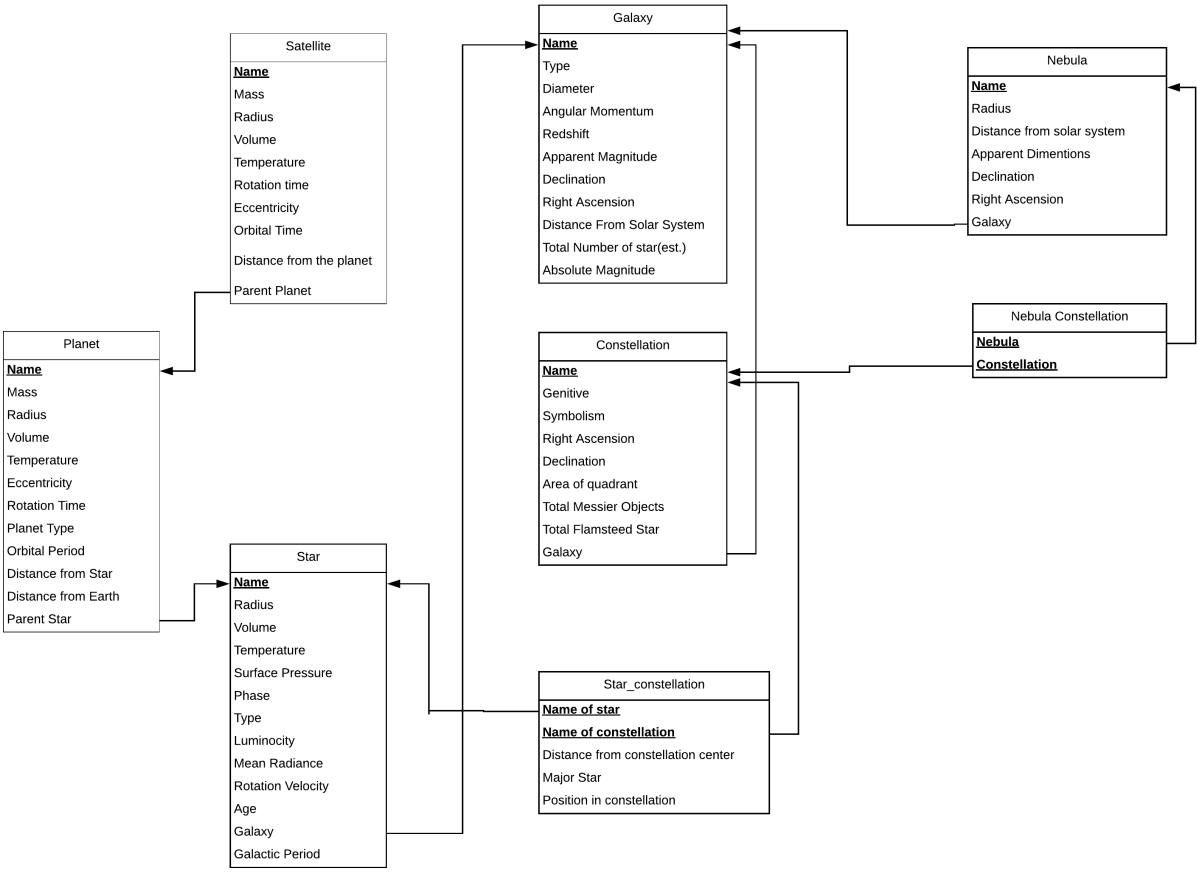

Fig: Schema Diagram of the Project

Assumptions:

1.Entity Relationship Model:

In the entity relationship model, star, planet, satellite, nebula, galaxy, constellation is regarded as entity. There are some relation between a satellite and planet, planet and star, star and galaxy, star and constellation, constellation and galaxy, constellation and nebula, nebula and galaxy.

There can be no satellite which has no planet. So total participation from the satellite side. A planet can has several satellite whether a satellite can has only one planet. So one to many relation from planet to satellite. Same things is applicable for star to galaxy, planet to star, constellation to galaxy and nebula to galaxy. The remaining two relation between nebula to constellation and star to constellation is many to one but it's not necessary for a star or a nebula to be included in a constellation. So, here it is one to one.

2.Schema Model:

In the schema model, there are less table than in the entity relationship model. This is because some of the relation in the erd have been merged in the schema diagram. The relations which have many to one cardinality and total participation from the many side, it can be merged with the many side. That's why about more than 5 table has been merged in the schema model. The foreign reference are given by the rule and the primary key is defined by examined that the key constraint will never be violated.

Environment of Implementation

Oracle-11g is an entry-level, small-footprint database based on the Oracle Database 11g. I have used oracle-11g for my database project. Oracle Database enables to store data, update and efficiently retrieve it, with a high degree of performance. It includes a software, database consisting of meta-data, server process, oracle instance with shared memory. Oracle 11g has very user friendly interface and it is easy to run script and SQL commands. Table can be modified inside the software without writing any script. All the triggers or function can be seen one by one and their code also. The another part is oracle-11g enables one to modify the main script after running it without again load it.

I worked on oracle-11g because it seemed easy to use rather than the others. Besides, oracle SQL can be run easily in the environment. Other application such as MySQL, PostgreSQL is not seemed so much sophisticated for making the project as they have complex functionality. There is another reason is that in the lab oracle-11g was used. So it was easy and less time consuming for me to use the oracle-11g. Otherwise, I need to accustomed with two environment if I used the other platform without not using the oracle-11g.

Application of the Database

1.Application Scope:

Show the name of the planet which has maximum rotation time and minimum rotation time.

SQL query:

```
SELECT a.name as maximum_time,b.name as minimum_time
FROM planet a,planet b
WHERE a.rotation_time=(select max(rotation_time) from planet)
    and
    b.rotation_time=(select min(rotation_time) from planet);
```

Output:

MAXIMUM_TIME	MINIMUM_TIME
VENUS	EARTH
VENUS	MARS

2 rows returned in 0.00 seconds

2. Application Scope:

Show the name of the star which is defined as major in any constellation.

Sql Query:

```
SELECT star_name
from star_contellation
where major_star='YES';
```

Output:

STAR_NAME
VEGA
DENEB
ANTARES

3.Scope:

Show the name of the galaxy which have the maximum star.

SQL query:

```
SELECT name
from galaxy
where total_number_of_stars=(select max(total_number_of_stars)
                             from galaxy);
```

Output:

NAME
TRIANGULUM

4.Scope:

Show the name of the star where the star isnt included in any constellation.

SQL query:

```
SELECT name
from star
where name not in(select star_name
                  from star_contellation);
```

Output:

NAME
ALPHA ANDROMEDAE
SUN
GAMMA VELORUM
CANOPUS
BETA CETI
ALPHA GRUIS
ACHERNAR
PROXIMA CENTAURAI
GAMMA LEONIS
ALNILAM
More than 10 rows available. Increase rows selector to view more rows.

5. Scope:

Show the name of the planet which has no satellite.

SQL Query:

```
SELECT name
FROM planet

minus
(
SELECT parent_planet
FROM satellite
);
```

Output:

NAME
MERCURY
PLANET A
VENUS

6. Scope:

Show the name of the satellite whose temperature is greater than at least one of the planet.

SQL query:

```
SELECT distinct s.name
FROM satellite s,planet p
WHERE s.temperature=(select min(temperature) from planet);
```

Output:

NAME
PHOBOS
EUROPA

7. Scope:

Show the name of the planet which have the most satellite.

SQL query:

```
with a as (SELECT p.name, count(s.name) as nos
          FROM planet p join satellite s on(p.name=s.parent_planet)
          GROUP BY p.name
         ),
    b as (select max(nos) mnos from a)

SELECT a.name
FROM a,b
where a.nos=b.mnos;
```

Output:

NAME
MARS
JUPITER
SATURN

8.Scope:

Show all the types of stars which is a major star in any constellation.

SQL query:

```
SELECT distinct s.star_type
FROM star s join star_contellation sc on(s.name=sc.star_name)
WHERE sc.major_star='YES';
```

Output:

STAR_TYPE
YELLOW DWARFS
ORANGE DWARF

9.Scope:

Show the number of all types of galaxy.

SQL query:

```
SELECT galaxy_type,count(galaxy_type)
from galaxy
GROUP BY galaxy_type;
```

Output:

GALAXY_TYPE	COUNT(GALAXY_TYPE)
BL LAC	1
RING	1
ELLIPTICAL	1
DWARF	1
STAR BURST	1
SPIRAL	6
RADIO	1
SPHEROIDAL	1
LENTICULAR	1

10.Scope:

Show the name of the planet which has maximum distance from its parent star.

SQL query:

```
SELECT name,Distance_from_parent_star
FROM planet
WHERE Distance_from_parent_star=(SELECT max(Distance_from_parent_star)
                                 FROM planet
                                );
```

Output:

NAME	DISTANCE_FROM_PARENT_STAR
SATURN	14340000000
PLANET A	14340000000

11. Scope:

Find luminocity of all stars using suns luminosity as a unit.

SQL query:

```
with a as(
    select luminosity as unit
    from star
    where name='SUN'
)
select name,round(luminosity/(select unit from a))
from star;
```

Output:

NAME	ROUND(LUMINOSITY/(SELECTUNITFROMA))
SUN	1
SIRIUS	1
VEGA	1
RIGEL	2
POLARIS	2
CANOPUS	2
ALTAIR	4
ARCTURUS	2
DENEB	4
BETELGEUSE	1
More than 10 rows available. Increase rows selector to view more rows.	

12. Scope:

Find the name of the nebula which isn’t included by any constellation.

SQL query:

```
select name
from nebula
minus
(
SELECT nebula_name
from nebula_constellation
)
```

Output:

NAME
LAGOON

13.Scope:

Show the pair of galaxy and constellation where galaxy has more declination than constellation.

SQL query:

```
SELECT g.name,c.name
from galaxy g join constellation c on (g.name=c.galaxy_name)
where g.declination>c.declination;
```

Output:

NAME	GALAXY_NAME
URSA MAJOR	MILKY WAY

14.Scope:

Show the name of the star which have the maximum planet

SQL query:

```
with a as (
select count(name) as ct_name from planet group by parent_star
)
select parent_star
from planet
group by parent_star
having count(name)=(select max(ct_name) from a);
```

Output:

PARENT_STAR
SUN

15. Scope:

Show the oldest star which is a major star in any constellation.

SQL query:

```
with a as(select age
from star s join star_contellation sc on (name=star_name))

select name,age
from star
where age=(select max(age) from a);
```

Output:

NAME	AGE
ANTARES	7700000000

16.Scope:

Show if a planet is livable or not liveable based on the temperature of the planet.

SQL query:

```
select Name,(case
            when temperature >=277 and temperature <297 then 'Liveable'
            else 'Not liveable'
          end
         ) as Result
from planet;
```

Output:

NAME	RESULT
EARTH	Liveable
VENUS	Not liveable
MERCURY	Not liveable
MARS	Not liveable
JUPITER	Not liveable
SATURN	Not liveable
PLANET A	Not liveable

Conclusions and Discussions:

Limitations of the Project:

The project is not fully done and has a lot of works left to do. Though it is planned to make this database a very realistic one, it was hard to to collect all the data about all the components and put them in the database correctly. It is also not certain that all the data will be authentic due to source limitation. There is a fatal limitation in the database. If a new star or nebula or constellation is discovered and it can't be sure that in which galaxy it is located then the star or nebula name can't be inserted into the database for parent key constraint. But in spite of being all of these limitation present in the project, these project is seemed very much helpful.

Future Possibilities of the project:

In future, the project can be upgraded and the limitations that are mentioned above can be eradicated by the required steps. Even, attribute of an entity can added easily by not violating any of the constraints. So, it can be said that the project is scalable.

Overall Discussion:

Though I can't say that I have completed the project , but still it taught me things in a very realistic way. I have learned many things about oracle SQL and it's application in varieties way. I could make this project more creative, more beautiful, more effective but some lack of knowledge and some lack of hard work, the project has so many flaws. This things will help me in future when I will work on another database project. Using different data type for different functionality and many alternative uses were learned during working on the project. The project was seemed to be very easy at first, but as the time goes, the project gradually turns into a tough one. As I never worked on a database project before, all of these was a new experience as well as new thrill. I got really some fun working on the project. If I would give more effort, the project would be better than it is now.