

**Database Design
of
Astronomy and Astrophysics C++ Library**

**Teacher : Dr. Saman Shishe Chi
Student : Ramtin Kosari**

QSSL





QSSL

```
# ifndef QSSL
/**
 * @file QSSL.hpp
 * @author Ramtin Kosari (ramtinkosari@gmail.com)
 * @brief QSSL Open Source Astronomy and Astrophysics Library
 * @note Enjoy Using this Library
 * @date 2024-01-06
 */
# define QSSL
# endif // QSSL
```

Table of Contents

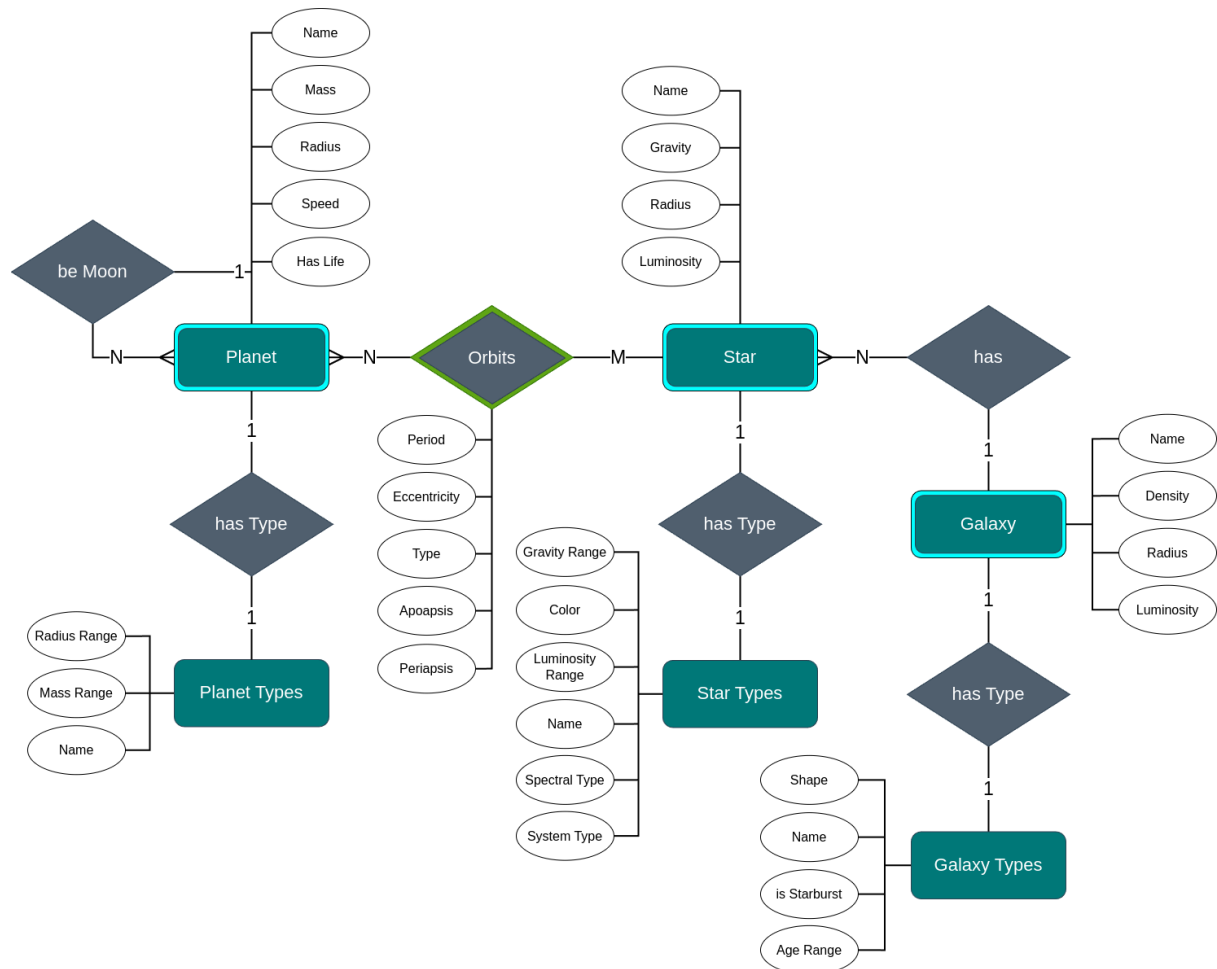
- **Overview**
- **ER Diagram**
 - ER Diagram
 - Details
 - Attributes
- **Tables**
 - Tables Diagram
 - Tables Queries
- **Example**
 - Example Queries

Overview

QSSL is a comprehensive open source C++ project that serves as an all-in-one toolset for astrophysicists and concerned developers. it covers a wide range of astronomical subjects including calculations of orbits, celestial trigonometry, celestial mechanics, astrophysics and cosmology also powerful tools for astronomical image processing based on artificial intelligence

ER Diagram

Here is Diagram of QSSL's Celestial Objects Database



in this Diagram there are 6 Tables (3 Weak Tables), 6 Relations, with Maximum Degree of 3, also It has 1 by 1, 1 by N and N by M Cardinalities.

Details

- **Entities**

- **Galaxy Types**

This table categorizes different types of galaxies.
Example types : Spiral, Elliptical, Irregular.

- **Galaxy**
This table represents individual galaxies.
- **Star Types**
This table categorizes different types of stars.
Example types : Main Sequence, Red Giant, White Dwarf.
- **Star**
This table represents individual stars.
Attributes are like Mass, Temperature, Luminosity, etc.
- **Planet Types**
This table categorizes different types of planets.
Example types: Terrestrial, Gas Giant, Dwarf Planet.
- **Planet**
This table represents individual planets.
Attributes like Radius, Mass, Orbital_Period, etc.

● **Relations**

- **Galaxy**
 1. 1 galaxy can only have 1 galaxy type and vise versa
 2. 1 galaxy can have N starts
- **Star**
 3. N stars can be in 1 galaxy
 4. 1 start can only have 1 star type and vise versa
 5. M stars can have N orbiting planets
- **Planet**
 6. N planet can be moon of 1 planet
 7. N planets can orbit around M stars
 8. 1 planet can only have 1 planet type

● **Attributes**

● **Planet Types**

- Type Name : like Rock or Gas planets
- Radius Range : range of planet radius
- Mass Range : range of planet mass

● **Planet**

- Name : name of planet
- Mass : mass of planet
- Radius : radius of planet
- Speed : speed of planet that is orbiting around another object
- has Life : if planet is habitable

- **Star Types**

- Gravity Range : range of stars gravity
- Color : color of stars in type
- Luminosity Range : range of star luminosity for type
- Type Name : name of star type
- Spectral Types : spectral type of star like G or O
- System Type : Binary System or 3 Body System

- **Star**

- Name : name of star
- Gravity : gravity of star
- Radius : radius of star
- Luminosity : luminosity of star

- **Galaxy Types**

- Shape : shape of galaxy like Spiral or Ellipse
- Name : type name
- is Starburst : types of starbursting galaxies
- Age Range : range of galaxy age in type

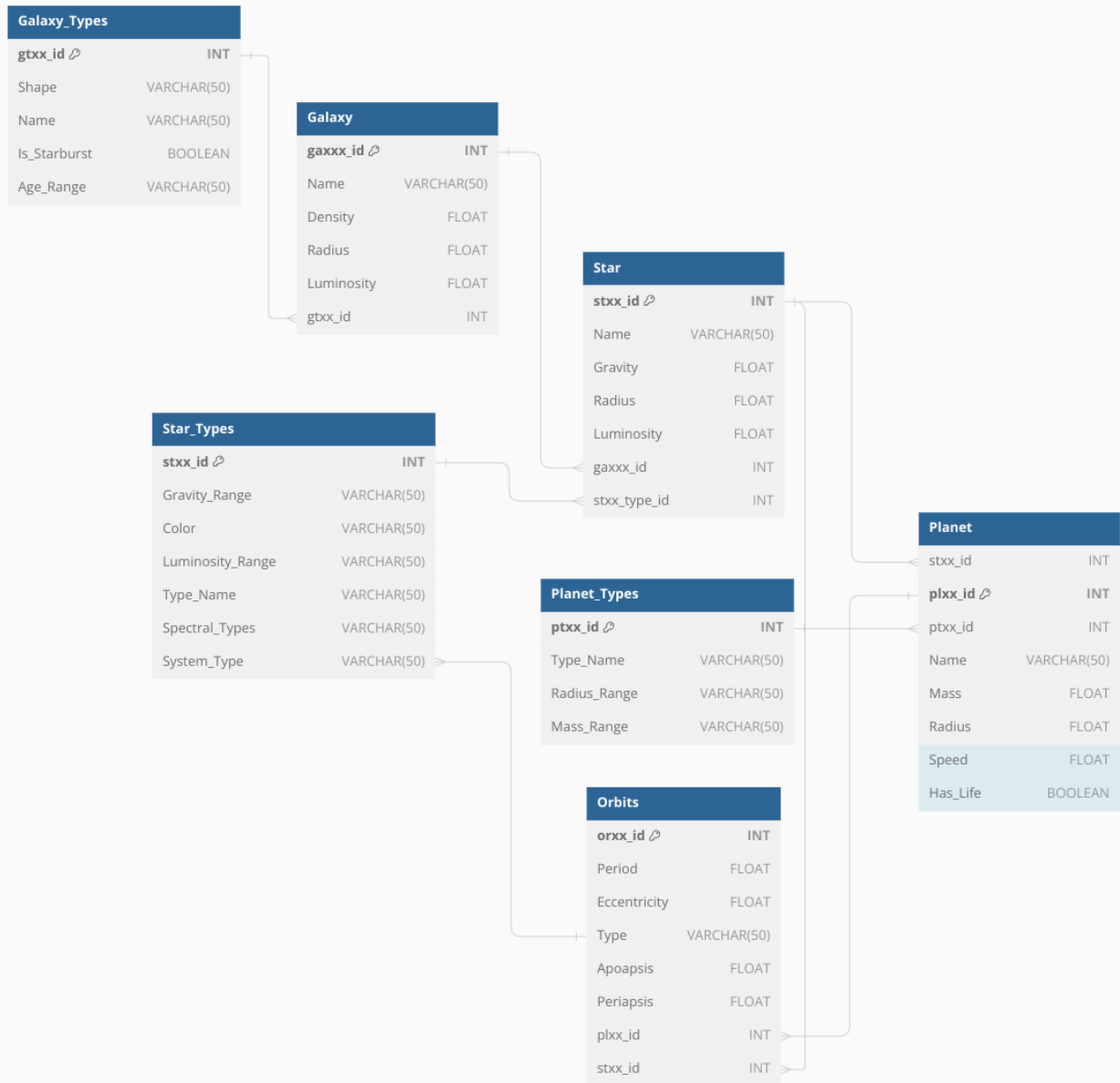
- **Galaxy**

- Name : name of galaxy
- Density : density of stars in galaxy
- Radius : radius of galaxy
- Luminosity : Luminosity of galaxy

- **Orbits**

- Period : period orbit
- Eccentricity : eccentricity of orbit
- Type : type of orbit
- Apoapsis : apoapsis of orbit
- Periapsis : periapsis of orbit

- **Tables Diagram**



- **Tables Queries**

- **Galaxy Types**

```
CREATE TABLE Galaxy_Types (  
    gtxx_id INT PRIMARY KEY,  
    Shape VARCHAR(50),  
    Name VARCHAR(50),  
    Is_Starcburst BOOLEAN,  
    Age_Range VARCHAR(50)  
);
```

- **Galaxy**

```
CREATE TABLE Galaxy (  
    gaxxx_id INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Density FLOAT,  
    Radius FLOAT,  
    Luminosity FLOAT,  
    gtxx_id INT,  
    FOREIGN KEY (gtxx_id) REFERENCES Galaxy_Types(gtxx_id)  
);
```

- **Star Types**

```
CREATE TABLE Star_Types (  
    stxx_id INT PRIMARY KEY,  
    Gravity_Range VARCHAR(50),  
    Color VARCHAR(50),  
    Luminosity_Range VARCHAR(50),  
    Type_Name VARCHAR(50),  
    Spectral_Types VARCHAR(50),  
    System_Type VARCHAR(50)  
);
```

- **Star**

```
CREATE TABLE Star (  
    stxx_id INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Gravity FLOAT,  
    Radius FLOAT,  
    Luminosity FLOAT,  
    gaxxx_id INT,  
    stxx_type_id INT,  
    FOREIGN KEY (gaxxx_id) REFERENCES Galaxy(gaxxx_id),  
    FOREIGN KEY (stxx_type_id) REFERENCES Star_Types(stxx_id)  
);
```

- **Planet Types**

```
CREATE TABLE Planet_Types (  
    ptxx_id INT PRIMARY KEY,  
    Type_Name VARCHAR(50),  
    Radius_Range VARCHAR(50),  
    Mass_Range VARCHAR(50)  
);
```

- **Planet**

```
CREATE TABLE Planet (  
    plxx_id INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Mass FLOAT,  
    Radius FLOAT,  
    Speed FLOAT,  
    Has_Life BOOLEAN,  
    ptxx_id INT,  
    FOREIGN KEY (ptxx_id) REFERENCES Planet_Types(ptxx_id)  
);
```

- **Orbits**

```
CREATE TABLE Orbits (  
    orxx_id INT PRIMARY KEY,  
    Period FLOAT,  
    Eccentricity FLOAT,  
    Type VARCHAR(50),  
    Apoapsis FLOAT,  
    Periapsis FLOAT,  
    plxx_id INT,  
    stxx_id INT,  
    FOREIGN KEY (plxx_id) REFERENCES Planet(plxx_id),  
    FOREIGN KEY (stxx_id) REFERENCES Star(stxx_id)  
);
```

- **Examples**

- Check habitable planets

```
SELECT Name FROM Planet WHERE Has_Life = TRUE;
```

- List of galaxies that their size is bigger than 10000 ly

```
SELECT Name, Radius FROM Galaxy WHERE Radius > 10000;
```

- List all galaxies with a density above a certain threshold and sort them by luminosity

```
SELECT Name, Density, Luminosity FROM Galaxy WHERE  
Density > 1.5 ORDER BY Luminosity DESC;
```