Week 9 Assessment

Josh Wainwright
UID:1079596

1 Parameters

- Generalisation Hierarchy Levels: 2
- M:N Relationships: 2
- Symetric Recursive Relationship: 1 : 1
- Multi-valued Attribute: 1

2 Astronomical Objects

This database describes the classification of a few astronomical objects, namely stars, planets and asteroids. Each of these objects has some aspects in common and so a generalisation hierarchy is used. Each type of object has a mass and an average distance that it lies from earth.

Stars are a type of object. In addition to the object attributes, they have a luminosity, as viewed from earth. They can also exist in a binary system where a star orbits at most one other star. Each star is made up of a number of elements; hydrogen, helium, etc; each of which has an atomic mass and number.

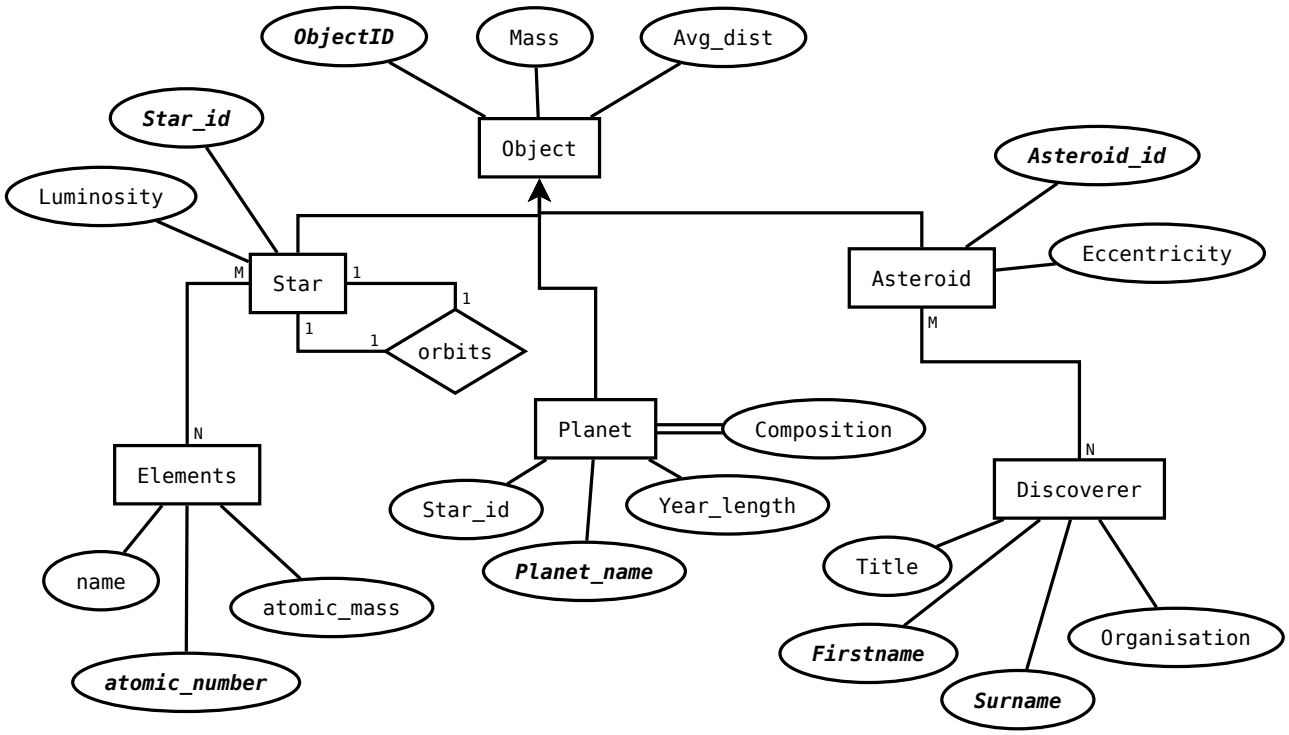
Planets exist in isolation, but contain information about their orbital duration, an optional star that they orbit (this must exist in the star table) and a list of constituents that they are made from.

Asteroids have an associated eccentricity of their orbit (how non-circular it is) as well as a number of discoverers who were responsible for finding and measuring it.

3 Business Rules

1. All stars, planets and asteroids are astronomical objects, called objects.
2. Stars can orbit zero or 1 other star.
3. Stars are composed of many elements and each element can appear in many stars.
4. Planets are composed of two or more elements.
5. Planet classification is either “rocky” or “gasseous”.
6. Asteroids all have one or more discoverers. Each asteroid can have multiple discoverers, representing a group discovery, and each discoverer can have discovered many asteroids.

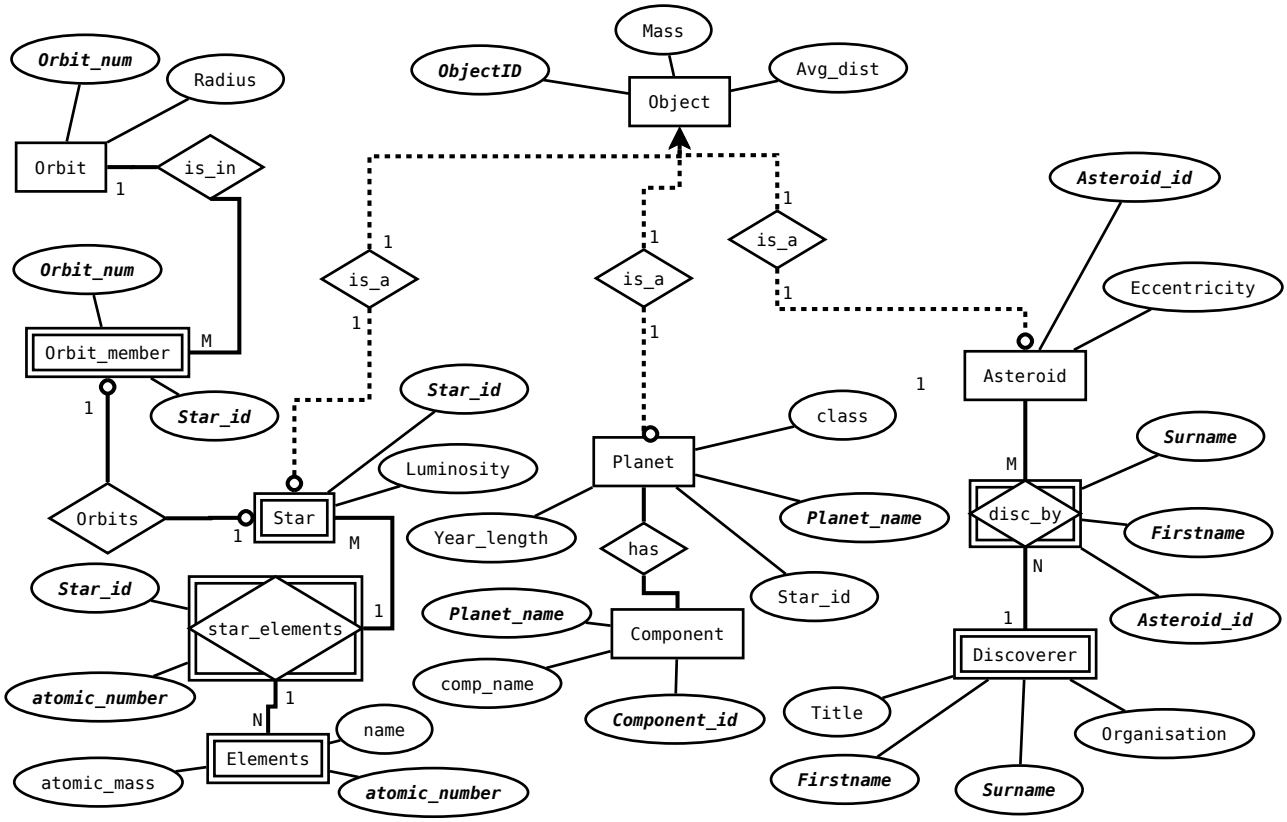
4 High Level ERD



4.1 Notes

- Chen Entity Relationship Diagram.
- The attributes making up the primary key is shown in bold.
- All generalisation hierarchies have exhaustive relationships.
- A subtype to supertype relationship is denoted with an arrow from subtype to supertype.
- Multi-valued attributes are shown with a double line.

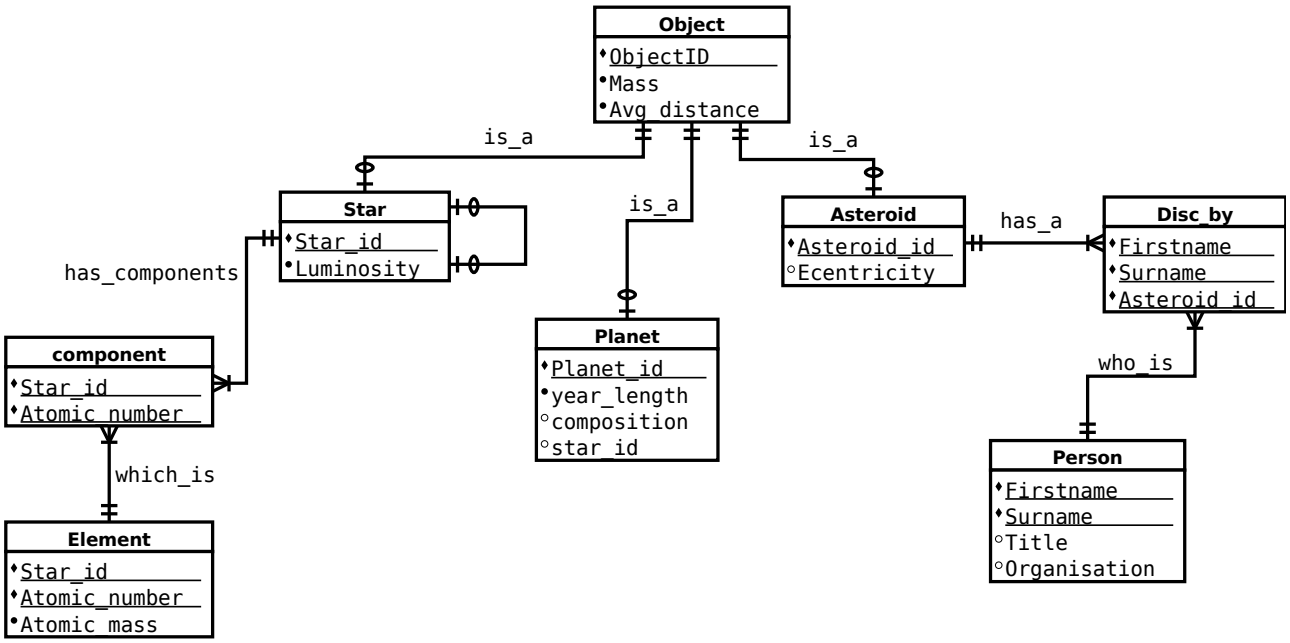
5 Low Level ERD



5.1 Notes

- Chen Entity Relationship Diagram.
- Attributes making up the primary key are shown in bold.
- Recursive relationships have been broken down.
- Weak relationships are shown with a dotted line, strong with a bold line.
- Weak entity types have a double border.
- Mulivalued attributes have been split into separate tables.

6 Low Level ERD



6.1 Notes

- Crows foot low level entity relationship diagram.

7 Table Creation

```
-- Super Type
CREATE TABLE Object (
        object_id SERIAL PRIMARY KEY,
        mass      INT    CHECK(mass > 0),
        avg_dist  INT    CHECK(avg_dist > 0)
);

-- Sub Type
CREATE TABLE Planet (
        object_id    SERIAL       REFERENCES Object,
        planet_name VARCHAR(30) UNIQUE PRIMARY KEY,
        year_length INT          CHECK(year_length > 0),
        class       CHAR(8)
);

-- Briding table for multivalued attribute
CREATE TABLE Component (
        planet_name  VARCHAR(30) REFERENCES Planet,
        component_id SERIAL,
        comp_name    VARCHAR(30),
        PRIMARY KEY (planet_name, component_id)
);

-- Sub Type
CREATE TABLE Asteroid (
        object_id     SERIAL REFERENCES Object,
        asteroid_id   SERIAL PRIMARY KEY,
        eccentricity DECIMAL CHECK(eccentricity > 0)
);

-- Table
CREATE TABLE Discoverer (
        firstname    VARCHAR(20),
        surname      VARCHAR(20) NOT NULL,
        title        VARCHAR(5)  NOT NULL,
        organisation VARCHAR(30),
        PRIMARY KEY (firstname, surname)
);

-- Briding table for M:N relationship
CREATE TABLE Disc_by (
        asteroid_id SERIAL       REFERENCES Asteroid,
        firstname   VARCHAR(20) REFERENCES Discoverer,
        surname     VARCHAR(20) REFERENCES Discoverer,
        date_disc   DATE,
        PRIMARY KEY (asteroid_id, firstname, surname)
);

-- Sub Type
CREATE TABLE Star (
        object_id  INT     REFERENCES Object,
        star_id    SERIAL PRIMARY KEY,
```

```
        luminosity INT     CHECK(luminosity > 0)
);

-- Table
CREATE TABLE Elements (
        atomic_number INT      CHECK(atomic_number > 0),
        atomic_mass   NUMERIC CHECK(atomic_mass > 0),
        PRIMARY KEY (atomic_number)
);

-- Briding table for M:N relationship
CREATE TABLE Star_Elements (
        star_id        SERIAL REFERENCES Star PRIMARY KEY,
        atomic_number INT     REFERENCES Elements
);

-- Bridging table for 1:1 symetric recursive relationship
CREATE TABLE orbit_member (
        orbit_num SERIAL,
        star_id   SERIAL REFERENCES Star,
        PRIMARY KEY (orbit_num, star_id)
);

-- Linking table for 1:1 symetric recursive relationship
CREATE TABLE orbit (
        orbit_num SERIAL  REFERENCES orbit_member PRIMARY KEY,
        radius    NUMERIC
);
```

8 Adding Entities

```
INSERT INTO Object VALUES
        (DEFAULT ,474, 2759),
        (DEFAULT ,204, 3679),
        (DEFAULT ,216, 1576),
        (DEFAULT ,601, 3916),
        (DEFAULT ,18, 4502),
        (DEFAULT ,744, 2993),
        (DEFAULT ,450, 10704),
        (DEFAULT ,162, 5063),
        (DEFAULT ,515, 2107),
        (DEFAULT ,315, 4107)
        ;

INSERT INTO star VALUES
        (1, DEFAULT, 4),
        (2, DEFAULT, 5),
        (3, DEFAULT, 6),
        (4, DEFAULT, 7)
        ;

INSERT INTO orbit VALUES
        (DEFAULT, 1000000),
```

```
        (DEFAULT, 92000000),
        (DEFAULT, 5100000),
        (DEFAULT, 25000000),
        (DEFAULT, 10000),
        ;

INSERT INTO orbit_member VALUES
        (1, 1),
        (1, 2),
        (2, 3),
        (2, 4)
        ;

INSERT INTO Star_Elements VALUES
        (1, 1),
        (1, 2),
        (1, 3),
        (1, 4),
        (1, 7),
        (1, 8),
        (1, 16),
        (2, 1),
        (2, 2),
        (2, 3),
        (3, 1),
        (3, 2),
        (3, 3),
        (3, 5),
        (3, 11),
        (3, 6)
        ;

INSERT INTO Elements VALUES
        (1, 1.001, 'Hydrogen'),
        (2, 4.003, 'Helium'),
        (3, 6.94, 'Lithium'),
        (4, 9.01, 'Beryllium'),
        (5, 10.81, 'Boron'),
        (6, 12.01, 'Carbon'),
        (7, 14.01, 'Nitrogen'),
        (8, 15.999, 'Oxygen'),
        (9, 18.998, 'Flourine'),
        (10, 20.18, 'Neon'),
        (11, 22.99, 'Sodium'),
        (12, 24.31, 'Magnesium'),
        (13, 26.98, 'Aluminium'),
        (14, 28.19, 'Silicon'),
        (15, 30.97, 'Phosphorus'),
        (16, 32.06, 'Sulphur'),
        (17, 35.45, 'Chlorine')
        ;

INSERT INTO planet VALUES
        (5, 'Zark', 1.3, 'rocky'),
        (6, 'Jupiter', 12, 'rocky'),
```

```
        (7, 'Io', 4.3, 'gasseous')
        ;

INSERT INTO component VALUES
        ('Zark', DEFAULT, 'plutonium'),
        ('Zark', DEFAULT, 'carbon'),
        ('Zark', DEFAULT, 'hydrogren'),
        ('Jupiter', DEFAULT, 'iron'),
        ('Jupiter', DEFAULT, 'silicon'),
        ('Jupiter', DEFAULT, 'oxygen'),
        ('Io', DEFAULT, 'iron'),
        ('Io', DEFAULT, 'lithium'),
        ('Io', DEFAULT, 'silicon'),
        ('Io', DEFAULT, 'carbon')
        ;

INSERT INTO asteroid VALUES
        (8, DEFAULT, 0.8),
        (9, DEFAULT, 0.1),
        (10, DEFAULT, 0.3)
        ;

INSERT INTO disc_by VALUES
        (1, 'Michael', 'Farrell', 20/06/1998),
        (2, 'Katherine', 'Pearson', 29/09/1991),
        (2, 'David', 'King', 29/09/1991),
        (2, 'Amelie', 'Fleming', 29/09/1991),
        (3, 'Kate', 'Slater', 04/04/1999),
        (3, 'Freddie', 'Gould', 04/04/1999),
        (3, 'Sophie', 'Freeman', 04/04/1999)
        ;

INSERT INTO discoverer VALUES
        ('Mr', 'Michael', 'Farrell', 'ESA'),
        ('Prof', 'Katherine', 'Pearson', 'NASA'),
        ('Prof', 'David', 'King', 'NASA'),
        ('Mrs', 'Amelie', 'Fleming', 'NASA'),
        ('Mrs', 'Kate', 'Slater', 'NASA'),
        ('Sir', 'Freddie', 'Gould',),
        ('Prof', 'Sophie', 'Freeman', 'NASA')
        ;
```

9 Queries

```
-- Get all the stars.
jaw097=> select * from star;
 object_id | star_id | luminosity
-----------+---------+------------
         1 |      21 |          4
         2 |      22 |          5
         3 |      23 |          6
         4 |      24 |          7
(4 rows)
```

```
-- Get all the planets.
jaw097=> SELECT * FROM planet
 object_id | planet_name | year_length |  class
-----------+-------------+-------------+---------
         5 | Zark        |           1 | rocky
         6 | Jupiter     |          12 | rocky
         7 | Io          |           4 | gasseous
(3 rows)

-- Get all the asteroids.
jaw097=> SELECT * FROM Asteroid;
 object_id | asteroid_id | eccentricity
-----------+-------------+--------------
         8 |          16 |          0.8
         9 |          17 |          0.1
        10 |          18 |          0.3
(3 rows)

-- Get all the components whether they are present in a planet or not.
jaw097=> SELECT * FROM component;
 planet_id | component id | comp name
-----------+--------------+-----------
         5 |           11 | iron
         5 |           12 | silicon
         5 |           13 | oxygen
         6 |           14 | iron
         6 |           15 | lithium
         6 |           16 | silicon
         6 |           17 | carbon
( 7 rows )

-- Get the details of all planets.
jaw097=> SELECT objectid, planetid, yearlength, class, mass
      -> FROM planet, object
      -> WHERE planet.object_id = object.objectid;
 object id | planet id | year length | class    | mass
-----------+-----------+-------------+----------+------
         5 |         1 |           1 | rocky    | 18
         6 |         2 |          12 | rocky    | 744
         7 |         3 |           4 | gasseous | 450

-- Get all the different components that are accounted for in at least
-- one planet.
jaw097=> SELECT DISTINCT (comp_name)
      -> FROM component, planet
      -> WHERE planet.planet_name = component.planet_name;
 comp name
-----------
 plutonium
 hydrogren
 iron
 silicon
 lithium
 oxygen
```

```
 carbon
( 7 rows )

-- Get the surname and organisation of the discoverers of any asteroid
-- with an eccentricity greater than 0.2.
jaw097=> SELECT discoverer.surname, discoverer.organisation
      -> FROM discoverer, disc_by, asteroid
      -> WHERE asteroid.asteroid_id = disc_by.asteroid_id AND
      ->       disc_by.firstname = asteroid.firstname AND
      ->       disc_by.surname = asteroid.surname AND
      ->       asteroid.eccentricity > 0.2;
 surname | organisation
---------+--------------
 Farrell | ESA
 Pearson | NASA
 King    | NASA
 Fleming | NASA
```