University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Monday, March 3rd, 2025

Group Number: 124

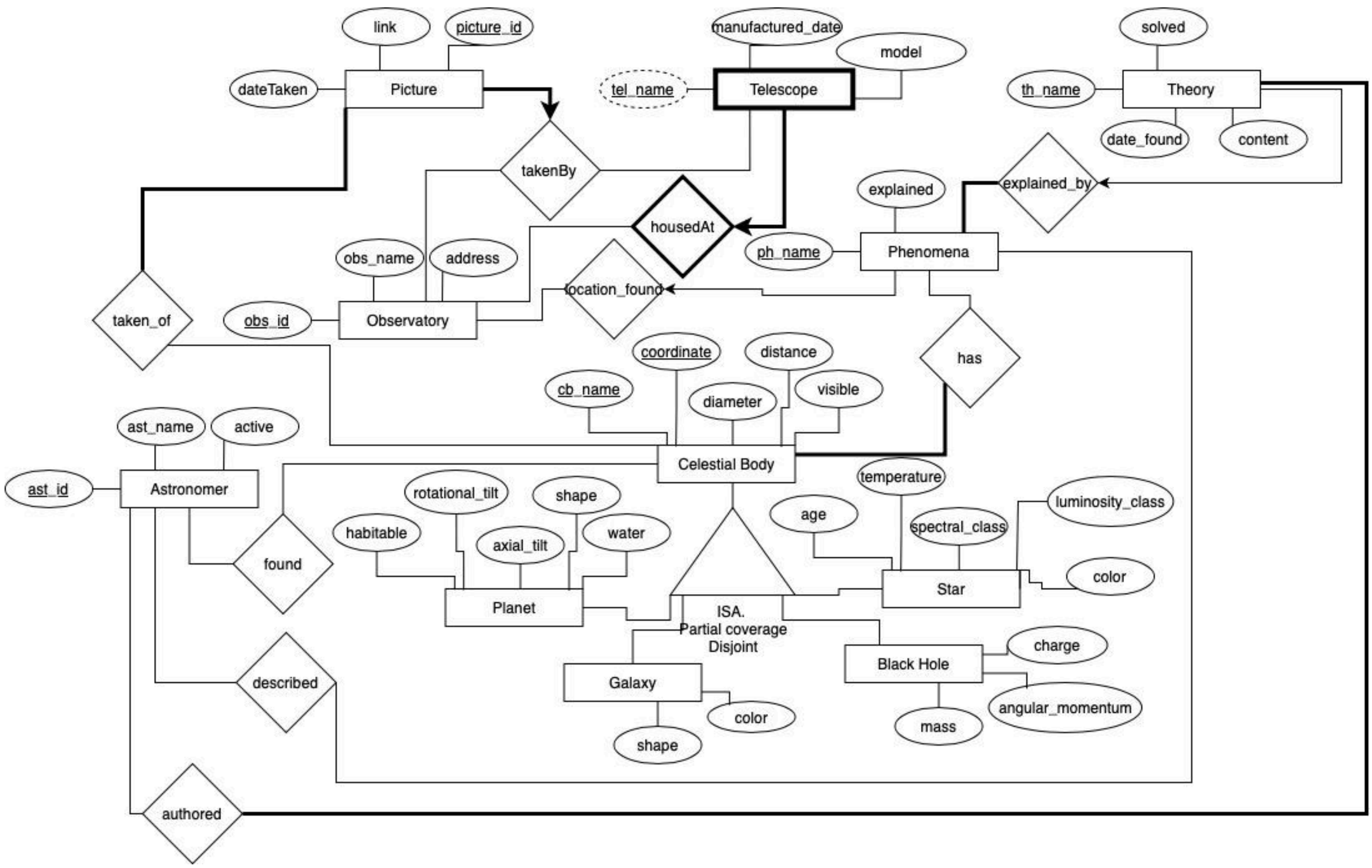
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
QueAnh Ngo	83827527	g4j1w	emeraldnqa@gmail.com
Senlin Sun	50218502	u0a8c	sun.senlin0@gmail.com
Raymond Li	58216474	x3c6y	li.raymond04@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Project summary

The domain of our project will be in space science and astronomy, which involves tracking multiple celestial bodies, phenomenas, theories, and images. The database will store attributes of different type of celestial bodies, phenomena, theories, astronomers, observatories, and telescopes. Beside storage, the application will also linking different entities together through multiple relationships such as found, authored, where and how the pictures are taken, or how different theories explain different phenomena, or tracking astronomers research work, for efficient access and retrievals.



3. ER Diagram and changes

Changes from Milestone 1 ER diagram:

- Making sure Telescope - Observatory weak entity relationship is shown on the ER diagram

- Picture entity has an extra primary key called picture_id since we realized that dateTaken is not a good primary key for picture, since multiple pictures can be taken on the same day, so they are not a good identifier for the identity, so we have added a new attribute called picture_id, and make it a primary key of the Picture entity
- Additionally, we have also decide to add coordinate to be a primary key of Celestial_Body as well, since coordinate are also a good identifier for Celestial_Body entity since they are unique, and help identify the tuples
- We have also use the appropriate vocabulary of the ISA hierarchies coverage. Specifically in Milestone 1, we used no coverage. We realized that the right vocabulary should be partial coverage to indicate the some of the Celestial_Body entity can stay the the parent class, or either one of the children class, which represent more of the real life situation where not all of the Celestial Body have to be either planet, star, galaxy or black hole, they can also be supernova, or asteroid.
- The naming of some of the relationship are also change in order to be consistent with the relational schema:
    - takenOf change to taken_of
    - locationFound change to location_found
    - explainedBy change to explained_by
- Renaming the primary key and attribute of the following entities for easier to differentiate these primary keys since initially they have very similar name, or some of the name change made for easier to read when working on the database:
    - Astronomer primary key from astID to ast_id,
    - Celestial_Body primary key from name to cb_name
    - Phenomena primary key from name to ph_name
    - Theory primary key from name to th_name
    - Telescope primary key from name to tel_name
    - Observatory primary key from obsID to obs_id
    - Observatory attribute from name to obs_name

4. Relational Schema

Note: Underline are primary keys, and bold are foreign keys

Celestial_Body: (cb_name: VARCHAR(40), coordinate: VARCHAR(100), visible: BOOLEAN, distance: DOUBLE PRECISION, diameter: DOUBLE PRECISION)

Star: (**cb_name:** VARCHAR(40), **coordinate**: VARCHAR(100), age: INTEGER, temperature: INTEGER, spectral_class: VARCHAR(40), luminosity_class: VARCHAR(40), color: VARCHAR(40))

Planet: (**cb_name:** VARCHAR(40), **coordinate**: VARCHAR(100), habitable: BOOLEAN, rotational_tilt: DOUBLE PRECISION, axial_tilt DOUBLE PRECISION, shape: VARCHAR(10), water: BOOLEAN)

Blackhole: (**cb_name:** VARCHAR(40), **coordinate:** VARCHAR(100), charge: DOUBLE PRECISION, angular_momentum: DOUBLE PRECISION, mass: DOUBLE PRECISION)

Galaxy:(**cb_name:** VARCHAR(40), **coordinate:** VARCHAR(100), shape: VARCHAR(10), color: VARCHAR(20))

Observatory: (obs_id: INTEGER, obs_name: VARCHAR(40), address: VARCHAR(100)): obs_name NOT NULL

Astronomer: (ast_id: INTEGER, ast_name: VARCHAR(40), active: BOOLEAN): ast_name are unique and not null

ph_location_found (ph_name:VARCHAR(50), **obs_id:** INTEGER, explained: BOOLEAN)

picture_taken_by (picture_id: INTEGER, **obs_id:** INTEGER, date_taken: DATE, link: VARCHAR(2048), **tel_name:** VARCHAR(40))

tel_housed_at (tel_name: VARCHAR(40), manufactured_date: DATE, model: VARCHAR(40), **obs_id:** INTEGER)

found (**ast_id:** INTEGER, **cb_name:** VARCHAR(40), **coordinate:** VARCHAR(100))

described(**ast_id:** INTEGER, **ph_name:** VARCHAR(40))

th_explained_by: (th_name: VARCHAR(40), **ph_name:** VARCHAR(40), date_found: DATE, content: VARCHAR(4000), solved: BOOLEAN)

authored: (**ast_id:** INTEGER**, th_name:** VARCHAR(40)**, ph_name:** VARCHAR(40))

has(**ph_name:** VARCHAR(40), **cb_name:** VARCHAR(40)**, coordinate:** VARCHAR(100))

taken_of(**picture_id:** INTEGER,**, cb_name:** VARCHAR(40)**, coordinate:** VARCHAR(100))

5. Functional Dependencies:

Note: underlined are primary keys

Celestial_Body: cb_name, coordinate -> visibility, distance, diameter

Star: cb_name, coordinate -> age, temperature, spectral_class, luminosity_class, color

temperature -> spectral_class, luminosity_class, color

Planet: cb_name, coordinate -> habitable, rotational_tilt, axial_tilt, shape, water

habitable -> water

Blackhole: cb_name, coordinate -> charge, angular_momentum, mass

Galaxy: cb_name, coordinate -> shape, color

Observatory: obs_id -> address, obs_name

obs_name -> address

address -> obs_name

Astronomer: ast_id, ast_name -> active

ph_location_found: ph_name -> explained, obs_id

picture_taken_by: picture_id -> obs_id, date_taken, link, tel_name

tel_name -> obs_id

tel_housed_at: tel_name, obs_id **->**manufactured_date, model

tel_name -> manufactured_date, model

tel_name -> obs_id

th_explained_by: th_name, ph_name -> date_found, content, solved

authored: only trivial FD

has: only trivial FD

found: only trivial FD

described: only trivial FD

taken_of: only trivial FD

6. Normalization:

All the other schemas either have trivial FD, or the FDs depend on primary keys except for Star, Planet, picture_take_by, and tel_housed at. The normalization processes are below:

Note: Underlined is key, and bold is foreign key

Star: (**cb_name**, **coordinate**, age, temperature, spectral_class, luminosity_class, color)

Through BCNF normalization we have two more schema as follow:

Star(**cb_name**, **coordinate**, age, **temperature**)

Star1(temperature, spectral_class, luminosity_class, color)

by FD: temperature -> spectral_class, luminosity_class, color

Planet: (**cb_name**, **coordinate**, habitable, rotational_tilt, axial_tilt, shape, water)

Through BCNF normalization we have the following tables:

Planet(**cb_name**, **coordinate**, **habitable**, rotational_tilt, axial_tilt, shape)

Planet1(habitable, water)

By FD: habitable -> water

Observatory(obs_id, obs_name, address)

Through BCNF normalization we have the following tables:

Observatory(obs_id, **obs_name**)

Observatory1(obs_name, address)

By FD: obs_name -> address, address -> obs_name

picture_taken_by(picture_id, **obs_id**, date_taken, link, **tel_name**)

Through BCNF normalization we have the following tables:

picture_taken_by(picture_id, date_taken, link, **tel_name**) picture_taken_by1(<u>tel_name</u>, obs_id)

By FD: tel_name -> obs_id

tel_housed_at(<u>tel_name</u>, manufactured_date, model, **<u>obs_id</u>**)

Through BCNF normalization we have the following tables:

tel_housed_at(**<u>tel_name</u>**, manufactured_date, model)

tel_housed_at1(<u>tel_name</u>, obs_id)

By FD: tel_name -> obs_id

Note: Since tel_housed_at1 and picture_take_by1 are the same table, we can combine them together for optimization: tel_housed_at1(<u>tel_name</u>, obs_id)

7. SQL DDL Statements:

Note: While writing the DDL in Oracle, we realize that it does not support ON UPDATE for FOREIGN KEYS clause, so we only consider the case for ON DELETE. Since there's no ON UPDATE in Oracle we assume that the keys cannot be changed.

```
CREATE TABLE celestial_body (

  cb_name VARCHAR(40),

  coordinate VARCHAR(100),

  visible BOOLEAN ,

  distance DOUBLE PRECISION,

  diameter DOUBLE PRECISION,

  PRIMARY KEY(cb_name, coordinate)

);

CREATE TABLE Star1 (

  temperature INTEGER,
```

```
  spectral_class VARCHAR(40),

  luminosity_class VARCHAR(40),

  color VARCHAR(40),

  PRIMARY KEY(temperature)

);

CREATE TABLE Star (

  cb_name VARCHAR(40),

  coordinate VARCHAR(100),

  age INTEGER,

  temperature INTEGER,

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

 ON DELETE CASCADE,

  PRIMARY KEY(cb_name, coordinate),

  FOREIGN KEY(temperature) REFERENCES Star1(temperature)

  ON DELETE NO ACTION

);
```

Note: Star1 is the normalized relation from Star, since we don’t want accidental deletion of temperature and all the other attributes in Star1 schema, we will leave delete as default, so any deletion of temperature in Star1 will be rejected.

Note: Since cb_name, and coordinate are the primary keys of Star that reference from celestial_body which is the parent class of Star, in order to make sure that every entity that appear in Star also appear in the parent class we use ON DELETE CASCADE (this is to avoid orphan).

```
CREATE TABLE Planet1 (
```

```
  habitable BOOLEAN,

  water BOOLEAN,

  PRIMARY KEY(habitable)

);

CREATE TABLE Planet (

  cb_name VARCHAR(40),

  coordinate VARCHAR(100),

  habitable BOOLEAN,

  rotational_tilt DOUBLE PRECISION,

  axial_tilt DOUBLE PRECISION,

 shape VARCHAR(10),

  PRIMARY KEY(cb_name, coordinate),

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

 ON DELETE CASCADE,

  FOREIGN KEY(habitable) REFERENCES Planet1(habitable)

 ON DELETE NO ACTION

);
```

Note: Planet1 is the normalized relation from Star, since we don't want accidental deletion of habitable and all the other attributes in Planet1 schema, we will leave delete as default, so any deletion of habitable in Planet1 will be rejected.

Note: Since cb_name, and coordinate are the primary keys of Planet that reference from celestial_body which is the parent class of Planet, in order to make sure that every entity that appear in Planet also appear in the parent class we use ON DELETE CASCADE (this is to avoid orphan).

```
CREATE TABLE Blackhole (

  cb_name VARCHAR(40),

  coordinate VARCHAR(100),

  charge DOUBLE PRECISION,

  angular_momentum DOUBLE PRECISION,

  mass DOUBLE PRECISION,

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

  ON DELETE CASCADE,

  PRIMARY KEY(cb_name, coordinate)

);
```

Note: Since cb_name, and coordinate are the primary keys of Blackhole that reference from celestial_body which is the parent class of Blackhole, in order to make sure that every entity that appear in Blackhole also appear in the parent class we use ON DELETE CASCADE (this is to avoid orphan).

```
CREATE TABLE Galaxy (

  cb_name VARCHAR(40),

  coordinate VARCHAR(100),

  shape VARCHAR(40),

  color  VARCHAR(40),

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

  ON DELETE CASCADE,

  PRIMARY KEY(cb_name, coordinate)

);
```

Note: Since cb_name, and coordinate are the primary keys of Galaxy that reference from celestial_body which is the parent class of Galaxy, in order to make sure that every entity that appear in Galaxy also appear in the parent class we use ON DELETE CASCADE (this is to avoid orphan).

```
CREATE TABLE Observatory1 (

  obs_name VARCHAR(40),

  address VARCHAR(100) NOT NULL,

  PRIMARY KEY(obs_name),

);

CREATE TABLE Observatory (

  obs_id INTEGER,

  obs_name VARCHAR(40) NOT NULL,

  FOREIGN KEY(obs_name)REFERENCES Observatory1(obs_name)

  ON DELETE CASCADE,

  PRIMARY KEY(obs_id),

);
```

Note: We use ON DELETE CASCADE since we want to avoid orphaning the Observatory table since its foreign key from Observatory1 represents one thing, the observatory. If one part of the observatory is deleted, then the entire thing should be deleted.

```
CREATE TABLE Astronomer (

  ast_id INTEGER,

  ast_name VARCHAR(40) NOT NULL,

  active BOOLEAN,

  PRIMARY KEY(ast_id),
```

```
  UNIQUE(ast_name)

);

CREATE TABLE ph_location_found (

  obs_id INTEGER,

  ph_name VARCHAR(50),

  explained BOOLEAN,

  PRIMARY KEY(ph_name),

  FOREIGN KEY(obs_id)REFERENCES Observatory(obs_id)

  ON DELETE SET NULL

);
```

Note: Since the Phenomena entity can exist without knowing the Observatory, we can set obs_id as Null if the referenced obs_id is deleted in Observatory. Additionally since the relationship between Observatory and Phenomenon are many-to-many so the obs_id can be set as Null on deletion.

```
CREATE TABLE tel_housed_at1(

  tel_name VARCHAR(40),

  obs_id INTEGER,

  PRIMARY KEY(tel_name),

  FOREIGN KEY(obs_id)REFERENCES Observatory(obs_id)

  ON DELETE CASCADE

);
```

Note: Since this is a weak entity set, it should be deleted when its owner entity is deleted.

```
CREATE TABLE tel_housed_at(
```

```
  tel_name VARCHAR(40),

  manufactured_date DATE,

  model VARCHAR(40),

  PRIMARY KEY(tel_name),

  FOREIGN KEY(tel_name)REFERENCES tel_housed_at1(tel_name)

 ON DELETE CASCADE

);

CREATE TABLE picture_taken_by(

  picture_id INTEGER,

  date_taken DATE,

  link VARCHAR(2048),

  tel_name VARCHAR(40),

  PRIMARY KEY(picture_id),

  FOREIGN KEY(tel_name)REFERENCES tel_housed_at1(tel_name)

  ON DELETE SET NULL

);
```

Note: We use ON DELETE SET NULL since we want to keep the information that the picture exists even if the telescope it was taken by is deleted.

```
CREATE TABLE found (

  ast_id INTEGER,

  cb_name VARCHAR(40),

  coordinate VARCHAR(100),
```

```
  FOREIGN KEY(ast_id) REFERENCES Astronomer(ast_id)

  ON DELETE CASCADE,

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

  ON DELETE CASCADE,

  PRIMARY KEY(ast_id, cb_name, coordinate)

);
```

Note: Since the referenced foreign keys ast_id, cb_name, and coordinate are also the primary keys of the table, so we want to make sure that if any entity does not longer exist in the referenced schema, then they cannot participate in the relation.

```
CREATE TABLE described (

  ast_id INTEGER,

  ph_name VARCHAR(40),

  FOREIGN KEY(ast_id) REFERENCES Astronomer(ast_id)

  ON DELETE CASCADE,

  FOREIGN KEY(ph_name) REFERENCES ph_location_found(ph_name)

  ON DELETE CASCADE,

  PRIMARY KEY(ast_id, ph_name)

);
```

Note: Since the referenced foreign keys ast_id, ph_name, are also the primary keys of the table, so we want to make sure that if any entity does not longer exist in the referenced schema, then they cannot participate in the relation.

```
CREATE TABLE th_explained_by (

  th_name VARCHAR(40),

  ph_name VARCHAR(40),
```

```
  date_found DATE,

  content VARCHAR(4000),

  solved BOOLEAN,

  FOREIGN KEY(ph_name) REFERENCES ph_location_found(ph_name)

  ON DELETE CASCADE,

  PRIMARY KEY(th_name, ph_name)

);
```

Note: Since the referenced foreign key ph_name, is also the primary key of the table, so we want to make sure that if any entity does not longer exist in the referenced schema, then they cannot participate in the relation.

```
CREATE TABLE Authored (

  ast_id INTEGER,

  th_name VARCHAR(40),

  ph_name VARCHAR(40),

  FOREIGN KEY(ast_id) REFERENCES Astronomer(ast_id)

  ON DELETE CASCADE,

  FOREIGN KEY(th_name, ph_name) REFERENCES th_explained_by(th_name, ph_name)

  ON DELETE CASCADE,

  PRIMARY KEY(ast_id, th_name, ph_name)

);
```

Note: Since the referenced foreign keys ast_id, th_name, ph_name, are also the primary keys of the table, so we want to make sure that if any entity does not longer exist in the referenced schema, then they cannot participate in the relation.

```
CREATE TABLE has(
```

```
  ph_name VARCHAR(40),

  coordinate VARCHAR(100),

  cb_name VARCHAR(40),

  PRIMARY KEY(ph_name, cb_name, coordinate),

  FOREIGN KEY(ph_name)REFERENCES ph_location_found(ph_name)

  ON DELETE CASCADE,

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

  ON DELETE CASCADE

);
```

Note: Since the referenced foreign keys ph_name, cb_name, coordinate, are also the primary keys of the table, so we want to make sure that if any entity does not longer exist in the referenced schema, then they cannot participate in the relation.

```
CREATE TABLE taken_of(

  picture_id INTEGER,

  coordinate VARCHAR(100),

  cb_name VARCHAR(40),

  PRIMARY KEY(picture_id, cb_name, coordinate),

  FOREIGN KEY(picture_id)REFERENCES picture_taken_by(picture_id)

  ON DELETE CASCADE,

  FOREIGN KEY(cb_name, coordinate)REFERENCES celestial_body(cb_name, coordinate)

  ON DELETE CASCADE

);
```

Note: Since the referenced foreign keys picture_id, cb_name, coordinate, are also the primary keys of the table, so we want to make sure that if any entity does not longer exist in the referenced schema, then they cannot participate in the relation.

8. INSERT Statement (populate each schema with at least 5 tuples)

INSERT INTO Celestial_Body(cb_name, coordinate, visible, distance, diameter) VALUES

(‘Planet1’, ‘RA 21h 44m, Dec -15degree 10’ ’, TRUE, 8.23e-6, 7.16e-10),

(‘Planet2’, ‘RA 06h 45m, Dec -16degree 43’ ’, TRUE, 8.6, 1.81e-6),

(‘Planet3’, ‘RA 00h 42m, Dec +41degree 16’ ’, TRUE, 2.537e6, 0.0233),

(‘Planet4’, ‘RA 19h 34m, Dec -22degree 19’ ’, FALSE, 6.26e-4, 2.52e-9),

(‘Planet5’, ‘RA 05h 55m, Dec +07degree 24’ ’, TRUE, 642.5, 1.31e-5),

(‘Supernova1’, ‘RA 05h 35m 27.6s, Dec -69degree 16’ 44’ ’, TRUE, 168000, NULL),

(‘Planet6’, ‘RA 19h 24m 09.5s, Dec +37degree 51’ 47’ ’, TRUE, 1400, 1.6),

(‘Blackhole1’, ‘RA 12h 30m 49.4s, Dec +12degree 23’ 28’ ’, FALSE, 53000000, NULL),

(‘Galaxy1’, ‘RA 03h 19m 50s, Dec -19degree 25’ 47’ ’, TRUE, 64000000, NULL),

(‘Asteroid1’, ‘RA 02h 58m 48s, Dec +34degree 40’ 44’ ’, TRUE, 0.002, 0.15),

(‘Star1’, ‘RA 14h 29m', 'Dec -62degree 41’ ’, NULL, NULL, NULL),

(‘Star2’, ‘RA 18h 36m, Dec +38degree 47’ ’, NULL, NULL, NULL),

(‘Star3’, ‘RA 14h 39m, Dec -60degree 50’ ‘, NULL, NULL, NULL),

(‘Star4’, ‘RA 17h 45m, Dec -29degree 00’ ’, NULL, NULL, NULL),

(‘Star5’, ‘RA 12h 30m, Dec +12degree 23’ ’, NULL, NULL, NULL),

(‘Galaxy1’, ‘RA 19h 58m, Dec +35degree 12’ ’, NULL, NULL, NULL),

(‘Galaxy2’, ‘RA 06h 20m, Dec -00degree 21’ ’, NULL, NULL, NULL),

(‘Galaxy4’, ‘RA 16h 55m, Dec -40degree 44’ ’, NULL, NULL, NULL), (‘Galaxy5’, ‘RA 17h 45m, Dec -29degree 00’ ’, NULL, NULL, NULL),

(‘Blackhole2’, ‘RA 12h 30m, Dec +12degree 23’ ’,NULL, NULL, NULL),

(‘Blackhole3’, ‘RA 01h 33m, Dec +30degree 39’ ’,NULL, NULL, NULL),

(‘Blackhole4’, ‘RA 03h 19m, Dec -18degree 32’ ’, NULL, NULL, NULL),

(‘Blackhole5’, ‘RA 10h 45m, Dec -03degree 44’ ’, TRUE, 0, 12756);

INSERT INTO Star(cb_name, coordinate, age, temperature) VALUES

(‘Star1’, ‘RA 14h 29m', 'Dec -62degree 41’ ’, 0.25, 9940),

(‘Star2’, ‘RA 18h 36m, Dec +38degree 47’ ’, 8.0, 3500),

(‘Star3’, ‘RA 14h 39m, Dec -60degree 50’ ’, 4.85, 3042),

(‘Star4’, ‘RA 17h 45m, Dec -29degree 00’ ’, 4.85, 3042),

(‘Star5’, ‘RA 12h 30m, Dec +12degree 23’ ’, 0.45, 9602);

INSERT INTO Star1(<u>temperature</u>, spectral_class, luminosity_class, color) VALUES

(9940, ‘A1’, ‘V’, ‘White’),

(3500, ‘M1’, ‘I’, ‘Red’),

(3042, ‘M5’, ‘V’, ‘Red’),

(9602, ‘A0’, ‘V’, ‘White’),

(5790, ‘G2’, ‘V’, ‘Yellow’);

INSERT INTO Blackhole (cb_name, coordinate, charge, angular_momentum, mass) VALUES

(‘Blackhole1’, ‘RA 12h 30m 49.4s, Dec +12degree 23′ 28’ ’, 0, 0.99, 6.5e9),

(‘Blackhole2’, ‘RA 12h 30m, Dec +12degree 23’ ’,0, 0.998, 14.8), (‘Blackhole3’, ‘RA 01h 33m, Dec +30degree 39’ ’,0, 0.99, 6.6),

(‘Blackhole4’, ‘RA 03h 19m, Dec -18degree 32’ ’, 0, 0.95, 7.0),

(‘Blackhole5’, ‘RA 10h 45m, Dec -03degree 44’ ’, 0, 0.93, 4.1e6);

INSERT INTO Galaxy(cb_name, coordinate, shape, color) VALUES

(‘Galaxy1’, ‘RA 19h 58m, Dec +35degree 12’ ’, ‘Spiral’, ‘Yellow-White’),

(‘Galaxy2’, ‘RA 06h 20m, Dec -00degree 21’ ’, ‘Spiral’, ‘Blue-White’),

(‘Galaxy4’, ‘RA 16h 55m, Dec -40degree 44’ ’, ‘Elliptical’, ‘Yellow’),

(‘Galaxy5’, ‘RA 17h 45m, Dec -29degree 00’ ’, ‘Spiral’, ‘Blue-White’),

INSERT INTO Planet(cb_name, coordinate, habitable, rotational_tilt, axial_tilt, shape) VALUES

(‘Planet1’, ‘RA 21h 44m, Dec -15degree 10’ ’, TRUE, 23.5, 23.5, ‘Spherical’),

(‘Planet2’, ‘RA 06h 45m, Dec -16degree 43’ ’, TRUE, 25.2, 25.2, ‘Oblate’),

(‘Planet3’, ‘RA 00h 42m, Dec +41degree 16’ ’, FALSE, 177.4, 177.4, ‘Spherical’),

(‘Planet4’, ‘RA 19h 34m, Dec -22degree 19’ ’, FALSE, 3.1, 3.1, ‘Oblate’),

(‘Planet5’, ‘RA 05h 55m, Dec +07degree 24’ ’,FALSE, 26.7, 26.7, ‘Oblate’);

INSERT INTO Planet1(habitable, water) VALUES

(TRUE, TRUE),

(FALSE, FALSE);

INSERT INTO Observatory(obs_id, obs_name) VALUES

(1, ‘Observatory1’),

(2, ‘Observatory2’),

(3, ‘Observatory3’),

(4, ‘Observatory4’),

(5, ‘Observatory5’);

INSERT INTO Observatory1(obs_name, address) VALUES

(‘Observatory1’, ‘Somewhere in space, 500km above Earth’),

(‘Observatory2’, ‘1234 Toronto, Ontario, Canada’),

(‘Observatory3’, ‘4567, Hawaii, USA’),

(‘Observatory4’, ‘Somewhere in space too, 1200km above Earth’),

(‘Observatory5’, ‘8910, Puerto Rico, USA’);

INSERT INTO Astronomer(ast_id, ast_name, active) VALUES

(1, ‘Astronomer1’, FALSE),

(2, ‘Astronomer2’, FALSE),

(3, ‘Astronomer3’, TRUE),

(4, ‘Astronomer4’, FALSE),

(5, ‘Astronomer5’, FALSE);

INSERT INTO ph_location_found(ph_name, obs_id, explained) VALUES

(‘Phenomena1’, 1, TRUE),

(‘Phenomena2’, 2, TRUE),

('Phenomena3', 3, TRUE),

('Phenomena4', 4, TRUE),

('Phenomena5', 5, FALSE);

INSERT INTO tel_housed_at1(tel_name,obs_id) VALUES

('Telescop1', 1),

('Telescop5', 2),

('Telescope2', 3),

('Telescope3', 4),

('Telescope4', 5);

INSERT INTO tel_housed_at(tel_name, manufactured_date, model) VALUES

('Telescop1', 1990-04-24, 'Space-based Reflecting Telescope'),

('Telescope2', 1998-05-05, 'Ground-based Optical Telescope'),

('Telescope3', 1999-07-23, 'Space-based X-ray Telescope'),

('Telescope4', 1963-11-01, 'Ground-based Radio Telescope')

('Telescop5', 1990-04-24, 'Space-based Reflecting Telescope');

INSERT INTO picture_taken_by(picture_id, date_taken, link, tel_name) VALUES

(1, 2016-07-08, 'https://dummy.com/image1.jpg', 'Telescop1'),

(2, 2016-07-08, 'https://dummy.com/image2.jpg', 'Telescope2'),

(3, 2016-07-08, 'https://dummy.com/image3.jpg', 'Telescope3'), (4, 2016-07-08, ‘https://dummy.com/image4.jpg’, ‘Telescope4’),

(5, 2016-07-08,’https://dummy.com/image5.jpg’, ‘Telescop5’);

INSERT INTO found (ast_id, cb_name, coordinate) VALUES

(1, ’Planet4’, ‘RA 19h 34m, Dec -22degree 19’ ’),

(2, ’Planet5', ‘RA 05h 55m, Dec +07degree 24’ ’),

(2, ‘Supernova1’, ‘RA 05h 35m 27.6s, Dec -69degree 16’ 44’ ’),

(3, ‘Planet6’, ‘RA 19h 24m 09.5s, Dec +37degree 51’ 47’ ’),

(4, ‘Blackhole1’, ‘RA 12h 30m 49.4s, Dec +12degree 23’ 28’ ’);

INSERT INTO described(ast_id, ph_name) VALUES

(1, ‘Phenomena1’),

(3, ‘Phenomena2’),

(4, ‘Phenomena5’),

(1, ‘Phenomena2’),

(2, ‘Phenomena1’);

INSERT INTO th_explained_by: (th_name, ph_name, date_found, content, solved) VALUES

(‘Theory1’, ‘Phenomena1’, 1966-06-06, ‘Theory1 prove Phenomena1’, TRUE),

(‘Theory2’, ‘Phenomena1’, 1980-07-07, ‘Theory2 prove Phenomena1’, FALSE),

(‘Theory3’, ‘Phenomena2’, 1970-08-05, ‘Theory3 prove Phenomena2’, FALSE),

(‘Theory4’, ‘Phenomena3’, 1900-09-04, ‘Theory4 prove Phenomena3’, TRUE), (‘Theory5’, ‘Phenomena4’, 1901-10-03, ‘Theory5 prove Phenomena4’, TRUE),

(‘Theory6’, ‘Phenomena5’, 1910-11-02, ‘Theory6 prove Phenomena5’, FALSE),

(‘Theory7’, ‘Phenomena5’, 1920-12-01, ‘Theory7 prove Phenomena5’, TRUE);

INSERT INTO authored(ast_id, th_name, ph_name) VALUES

(1,‘Theory1’, ‘Phenomena1’),

(1,‘Theory2’, ‘Phenomena1’),

(2,‘Theory3’, ‘Phenomena2’),

(3,‘Theory4’, ‘Phenomena3’),

(4,‘Theory5’, ‘Phenomena4’),

(5,‘Theory6’, ‘Phenomena5’),

(5,‘Theory7’, ‘Phenomena5’);

INSERT INTO Has(ph_name, cb_name, coordinate) VALUES

(‘Phenomena1’, ‘Planet1’, ‘RA 21h 44m, Dec -15degree 10’ ’),

(‘Phenomena1’, ‘Planet2’, ‘RA 06h 45m, Dec -16degree 43’ ’),

(‘Phenomena1’, ‘Planet3’, ‘RA 00h 42m, Dec +41degree 16’ ’),

(‘Phenomena1’, ‘Planet4’, ‘RA 19h 34m, Dec -22degree 19’ ’),

(‘Phenomena1’, ‘Planet5’, ‘RA 05h 55m, Dec +07degree 24’ ’),

(‘Phenomena2’, ‘Supernova1’, ‘RA 05h 35m 27.6s, Dec -69degree 16’ 44’ ’),

(‘Phenomena1’, ‘Planet6’, ‘RA 19h 24m 09.5s, Dec +37degree 51’ 47’ ’),

(‘Phenomena3’, ‘Blackhole1’, ‘RA 12h 30m 49.4s, Dec +12degree 23’ 28’ ’), (‘Phenomena4’, ‘Galaxy1’, ‘RA 03h 19m 50s, Dec -19degree 25’ 47’ ’),

(‘Phenomena4’, ‘Asteroid1’, ‘RA 02h 58m 48s, Dec +34degree 40’ 44’ ’),

(‘Phenomena5’, ‘Star1’, ‘RA 14h 29m', 'Dec -62degree 41’ ’),

(‘Phenomena5’, ‘Star2’, ‘RA 18h 36m, Dec +38degree 47’ ’),

(‘Phenomena5’, ‘Star3’, ‘RA 14h 39m, Dec -60degree 50’ ’),

(‘Phenomena5’, ‘Star4’, ‘RA 17h 45m, Dec -29degree 00’ ’),

(‘Phenomena5’, ‘Star5’, ‘RA 12h 30m, Dec +12degree 23’ ’),

(‘Phenomena4’, ‘Galaxy1’, ‘RA 19h 58m, Dec +35degree 12’ ’),

(‘Phenomena4’, ‘Galaxy2’, ‘RA 06h 20m, Dec -00degree 21’ ’),

(‘Phenomena4’, ‘Galaxy4’, ‘RA 16h 55m, Dec -40degree 44’ ’),

(‘Phenomena4’, ‘Galaxy5’, ‘RA 17h 45m, Dec -29degree 00’ ’),

(‘Phenomena3’, ‘Blackhole2’, ‘RA 12h 30m, Dec +12degree 23’ ’),

(‘Phenomena3’, ‘Blackhole3’, ‘RA 01h 33m, Dec +30degree 39’ ’),

(‘Phenomena3’, ‘Blackhole4’, ‘RA 03h 19m, Dec -18degree 32’ ’),

(‘Phenomena3’, ‘Blackhole5’, ‘RA 10h 45m, Dec -03degree 44’ ’);

INSERT INTO taken_of(picture_id, cb_name, coordinate) VALUES

(1,’Planet3’, ‘RA 00h 42m, Dec +41degree 16’ ’),

(2,’Planet4’, ‘RA 19h 34m, Dec -22degree 19’ ’),

(3, ‘Planet5’, ‘RA 05h 55m, Dec +07degree 24’ ’),

(4, ‘Supernova1’, ‘RA 05h 35m 27.6s, Dec -69degree 16’ 44’ ’),

(5, ‘Planet6’, ‘RA 19h 24m 09.5s, Dec +37degree 51’ 47’ ’);

9. AI acknowledgement:

ChatGPT was used to generate random value for certain attributes in Celestial_Body schema. Prompts that we've used:

Give me examples of temperature, shape of galaxy in space, color of galaxies in space, distance and diameters.

Give me examples of luminosity class, spectral class, color and their related temperature

For coordinate of Celestial_Body is a primary key, the value need to be unique, we asked ChatGPT to generate 22 unique coordinates of celestial bodies in space, with the following prompts: Give 22 unique coordinates of celestial bodies in space