***Universal Table Normalization for Astronomy Data Management System***

***Step 1: Create a Universal (Denormalized) Table***

UNIVERSAL\_TABLE(

user\_id, user\_name, user\_email, user\_role, user\_password,

celestial\_object\_id, celestial\_object\_name, celestial\_object\_type,

celestial\_object\_discovery\_date, celestial\_object\_distance\_ly,

star\_id, star\_name, star\_spectral\_type, star\_temperature, star\_luminosity,

galaxy\_id, galaxy\_name, galaxy\_type, galaxy\_redshift, galaxy\_mass, galaxy\_distance\_ly,

asteroid\_id, asteroid\_name, asteroid\_diameter, asteroid\_composition, asteroid\_orbit\_type,

exoplanet\_id, exoplanet\_name, exoplanet\_host\_star\_id, exoplanet\_orbital\_period,

exoplanet\_mass, exoplanet\_radius, exoplanet\_atmosphere,

spectral\_data\_id, spectral\_data\_object\_id, spectral\_data\_type, spectral\_data\_wavelength\_nm,

observation\_log\_id, observation\_log\_user\_id, observation\_log\_object\_id,

observation\_log\_telescope, observation\_log\_date\_observed

)

***Step 2: Reduce to First Normal Form (1NF)***

**Problems in Universal Table:**

1. Contains multiple entity types in one table
2. Has repeating groups (multiple IDs for different entity types)
3. Contains redundant data

**1NF Requirements:**

* Each table cell should contain atomic values
* Each record needs to be unique
* No repeating groups

**Solution:**  
Break into separate tables based on entity types:

USERS(

user\_id (PK),

user\_name,

user\_email,

user\_role,

user\_password

)

CELESTIAL\_OBJECTS(

object\_id (PK),

name,

type,

discovery\_date,

distance\_ly

)

STARS(

star\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

spectral\_type,

temperature,

luminosity

)

GALAXIES(

galaxy\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

type,

redshift,

mass

)

ASTEROIDS(

asteroid\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

diameter,

composition,

orbit\_type

)

EXOPLANETS(

exoplanet\_id (PK),

name,

host\_star\_id (FK to STARS),

orbital\_period,

mass,

radius,

atmosphere

)

SPECTRAL\_DATA(

spectra\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

spectral\_type,

wavelength\_nm

)

OBSERVATION\_LOGS(

log\_id (PK),

user\_id (FK to USERS),

object\_id (FK to CELESTIAL\_OBJECTS),

telescope,

date\_observed

)

***Step 3: Reduce to Second Normal Form (2NF)***

**2NF Requirements:**

* Must be in 1NF
* All non-key attributes must depend on the entire primary key
* No partial dependencies

**Analysis:**

1. In the initial STARS table, if we had both star\_id and object\_id as composite PK, 'name' would be a partial dependency (depends only on object\_id)

**Solution:**

* Remove redundant attributes (like 'name' from subclass tables that already exist in CELESTIAL\_OBJECTS)
* Ensure all non-key attributes depend on the entire PK

***Updated Tables (2NF compliant):***

USERS (unchanged)

CELESTIAL\_OBJECTS (unchanged)

STARS(

star\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

spectral\_type,

temperature,

luminosity

) [removed 'name']

GALAXIES(

galaxy\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

type,

redshift,

mass

) [removed 'name' and 'distance\_ly']

ASTEROIDS(

asteroid\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

diameter,

composition,

orbit\_type

) [removed 'name']

EXOPLANETS (unchanged)

SPECTRAL\_DATA (unchanged)

OBSERVATION\_LOGS (unchanged)

***Step 4: Reduce to Third Normal Form (3NF)***

**3NF Requirements:**

* Must be in 2NF
* No transitive dependencies (non-key attributes shouldn't depend on other non-key attributes)

**Analysis:**

1. In EXOPLANETS: If we could calculate radius from mass and density, this would be a transitive dependency
2. In STARS: If luminosity could be calculated from temperature, this would be a transitive dependency

**Solution:**

* Remove derived attributes or document that they're intentionally denormalized for performance
* In this case, we'll assume these are base measurements, not derived values

**Final 3NF Tables:**

The tables from 2NF already satisfy 3NF requirements as there are no transitive dependencies in the current design.

***Additional Optimization: Boyce-Codd Normal Form (BCNF)***

**BCNF Requirements:**

* Must be in 3NF
* For any non-trivial dependency X → Y, X must be a superkey

**Analysis:**

* All tables satisfy BCNF as:
  + All non-key attributes depend only on the primary key
  + No non-trivial dependencies exist where the determinant isn't a candidate key

***Final Normalized Schema***

USERS(

user\_id (PK),

name,

email (UQ),

role,

password\_hash

)

CELESTIAL\_OBJECTS(

object\_id (PK),

name,

type,

discovery\_date,

distance\_ly

)

STARS(

star\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS, UQ),

spectral\_type,

temperature,

luminosity

)

GALAXIES(

galaxy\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS, UQ),

type,

redshift,

mass

)

ASTEROIDS(

asteroid\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS, UQ),

diameter,

composition,

orbit\_type

)

EXOPLANETS(

exoplanet\_id (PK),

name,

host\_star\_id (FK to STARS),

orbital\_period,

mass,

radius,

atmosphere

)

SPECTRAL\_DATA(

spectra\_id (PK),

object\_id (FK to CELESTIAL\_OBJECTS),

spectral\_type,

wavelength\_nm

)

OBSERVATION\_LOGS(

log\_id (PK),

user\_id (FK to USERS),

object\_id (FK to CELESTIAL\_OBJECTS),

telescope,

date\_observed,

notes

)

***Key Improvements in Normalized Design:***

1. **Eliminated redundancy**: Removed duplicate 'name' fields from subclass tables
2. **Proper inheritance**: Implemented superclass/subclass relationship via foreign keys
3. **Atomic values**: Each field contains only atomic values
4. **No partial dependencies**: All non-key attributes depend on the full primary key
5. **No transitive dependencies**: No non-key attributes depend on other non-key attributes
6. **BCNF compliance**: Every determinant is a candidate key