

Khyla King

CIS 276DB

Spring Semester

SQL Astrology Capstone

Astrology and Zodiac Database

The Zodiac & Astrology Database is a relational database project designed to explore the alignment between celestial symbolism and structured data. Built in Microsoft SQL Server Management Studio (SSMS), this system models core astrological concepts such as zodiac signs, planetary rulers, astrological houses, and compatibility while connecting them to fictional user profiles in a way that mimics real-world application logic.

This database captures not only technical best practices (normalization, foreign key constraints, reusable logic) but also a symbolic, metaphysical theme to show the abstract use of SQL. Astrology is used as a lens through which to understand relational structures, compatibility, and identity ultimately merging data science with cosmic design.

Technologies & Features

This project was developed using Microsoft SQL Server Management Studio (SSMS) utilizing T-SQL capabilities.

Key Features Include:

- **◆ Five normalized, relational tables:**
 - ZodiacSigns: Core sign data (modality, element, ruling planet)
 - Planets: Planetary names and meanings
 - Houses: Astrological house meanings, linked to signs
 - UserProfiles: Fictional users with birthdates and sign links
 - Compatibility: Relationship strengths between signs
- **◆ Primary and Foreign Key Constraints**

- ♦ **24 User Profiles** (2 for each zodiac sign) with date ranges to reflect proper astrological spans for users 18 and older for any compatibility matches
- ♦ **60+ total records** across all tables
- ♦ **View:** vw_UserAgeProfile — displays joined user data with calculated age and zodiac info
- ♦ **Function:** fn_GetZodiacSignFromBirthdate — returns zodiac sign based on a given date
- ♦ **Stored Procedures:**
 - sp_GetZodiacProfileByUserID: Returns detailed user zodiac info
 - sp_CheckUserCompatibility: Compares two users based on zodiac sign
 - sp_CheckZodiacCompatibility: Returns compatibility level between any two signs

Entity Relationship Diagram (ERD)

- An Entity Relationship Diagram (ERD) was created using SSMS's built-in diagramming tool and exported as an image. It visually represents all tables, primary keys, and foreign key relationships. The ERD image is included in the project folder as:
ERD_ZodiacDatabase_KhylaKing.png

To run this project from start to finish, the following steps should be followed within SSMS:

1. **Execute the table creation and insert scripts** in the following order:
 - 01_CreateTablesZodiacSigns.sql
 - 02_InsertData_ZodiacSigns.sql
 - 03_CreateTable_Planets.sql
 - 04_InsertData_Planets.sql
 - 05_CreateTable_Houses.sql
 - 06_InsertData_Houses.sql
 - 07_CreateTable_UserProfile
 - 08_InsertData_UserProfiles.sql
 - 09_CreateTable_Compatibility
 - 10_InsertData_Compatibility.sql
2. **Create views, functions, and procedures:**
 - 11_CreateViews.sql
 - 12_CreateFunctions.sql
 - 13_CreateStoredProcedures.sql

3. **Run test queries** using:

- 14_TestQueries.sql to confirm that all logic is functional
- 15_UserCountPerSign
- 16_UserCountBySign
- 17_UserInfoAndAgeCategory
- 18_ErrorMessage
- 19_Summary Query

By now all tables should have their information, with views, functions, procedures, and tests all successfully running with even a full display of the information for the Zodiac Profiles.