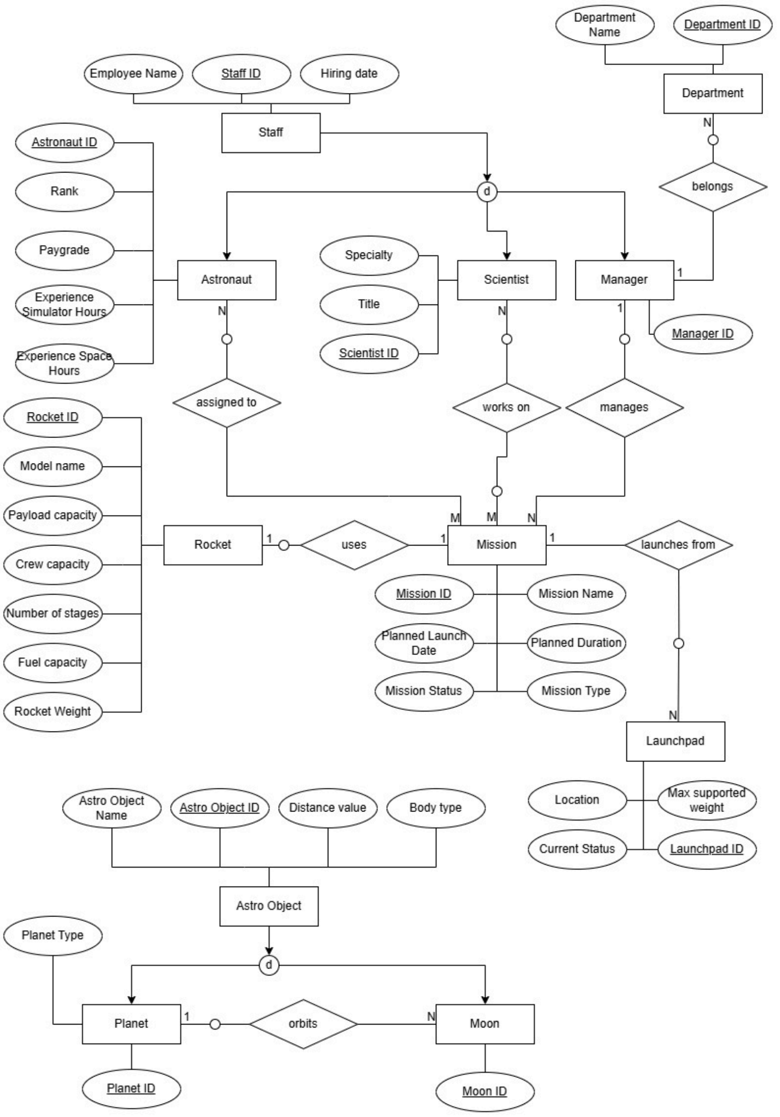
Assignment 2

E/R Diagram



Note that the E/R diagram follows a conceptual structure, and is strictly developed from the interviews provided. The implementation follows some practical additions in addition to this conceptual E/R diagram, which is marked with comments. If there are some differences from the E/R to the setup schema (like role descriptions for astronaut and scientist), this is why.

Assumptions

First and foremost, i have *added ID's to all of the entities* instead of complex composites of values in the database. This makes for a more simple construction and identification. Then, i assumed that *staff is a supertype* to the different types of employee, namely Astronaut, Scientist and Manager. This was to group the common values like ID and name in a single entity instead of spreading them on three separate entities. This implementation would make the ID's (primary keys) of the subtypes a foreign key of the supertype.

I assumed a *connection between a mission and it target body* (Astro Object), as i determined that a mission without a target would be illogical to not register.

For implementation, i assumed data types based on common practices. There are no explicit references to data types in the interviews, which makes all of them implicit assumptions.

Lastly, assumptions on the designated role of a scientist or astronaut during a mission was assumed. This was made to add some more context to missions, if they were to be examined at a later point. They are nullable if needed.

Constraints

Every table includes a PRIMARY KEY constraint for unique record identification. NOT NULL constraints were applied to fields whose values are deemed essential for a record's integrity (e.g., HiringDate, MissionName). In generalization hierarchies, the primary key of a sub-type (e.g., ManagerID in Manager) also functions as a foreign key to its super-type (StaffID in Staff).

Many FOREIGN KEY constraints are implemented to enforce referential integrity between tables. The central Mission table includes mandatory foreign keys to Launchpad, Rocket, Manager, and AstroObject, emphasizing the mission's dependency on these entities being present.

CHECKS *h*ave been made to ensure binary decisions (e.g. 'planet' or 'gas giant').

COMPOSITE KEYS of the ID's of both parties is used in the junction tables, which emphasizes the need for both parties to be present in order to be in the table.

Referential integrity actions were applied to foreign key constraints to manage data integrity. ON UPDATE CASCADE was applied, which meant that if a primary key in a parent table is updated, the corresponding foreign key values in child tables are automatically updated. ON DELETE NO ACTION was the primary choice for deletion in most mandatory 1:N relationships and for both sides of the junction tables. This explicit choice prevents the deletion of a parent record if child records still reference it. ON DELETE CASCADE was used for generalization hierarchies where the child record essentially is the parent record (e.g., manager/staff).

Implicit Mandatory/Optional participation

In addition to the explicitly stated participations (e.g., a mission must have a manager), i decided that a *single manager can only have one department*, mainly to avoid complexity. This means implementing a 1-N relationship instead of N-M. Lastly, i assumed that a rocket and a launchpad was mandatory for a mission to progress.