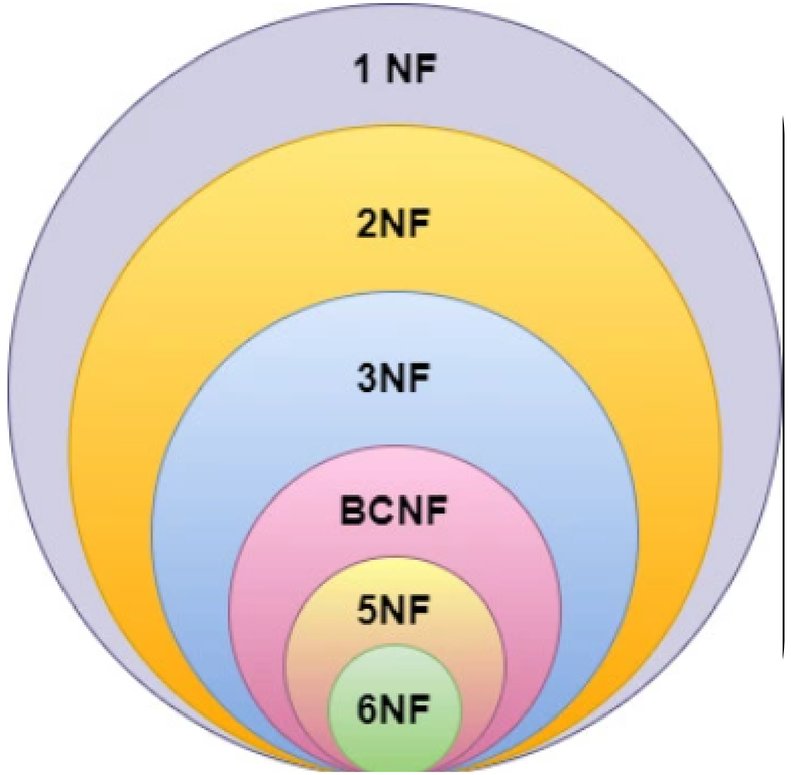
Performing Normalization to Employee Database

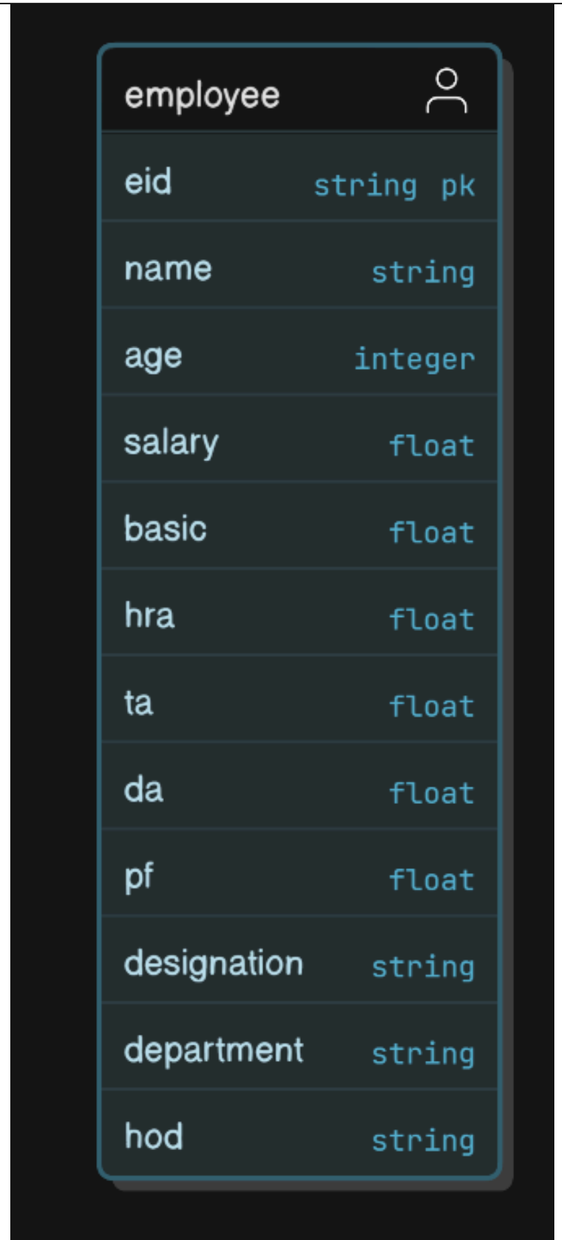
Akash Singh

Various Normalization Techniques Used

- Normalization is the process of organizing data to reduce redundancy and improve integrity.
- Levels of Normalization: 1NF, 2NF, 3NF, BCNF.



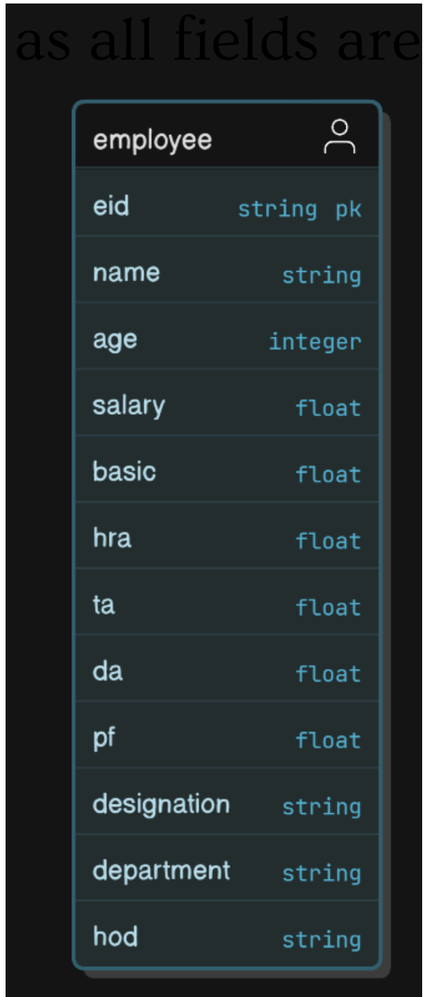
Basic Structire Before Normalizing



1NF

A table is in 1NF if it contains only atomic (indivisible) values and each column contains values of a single type.

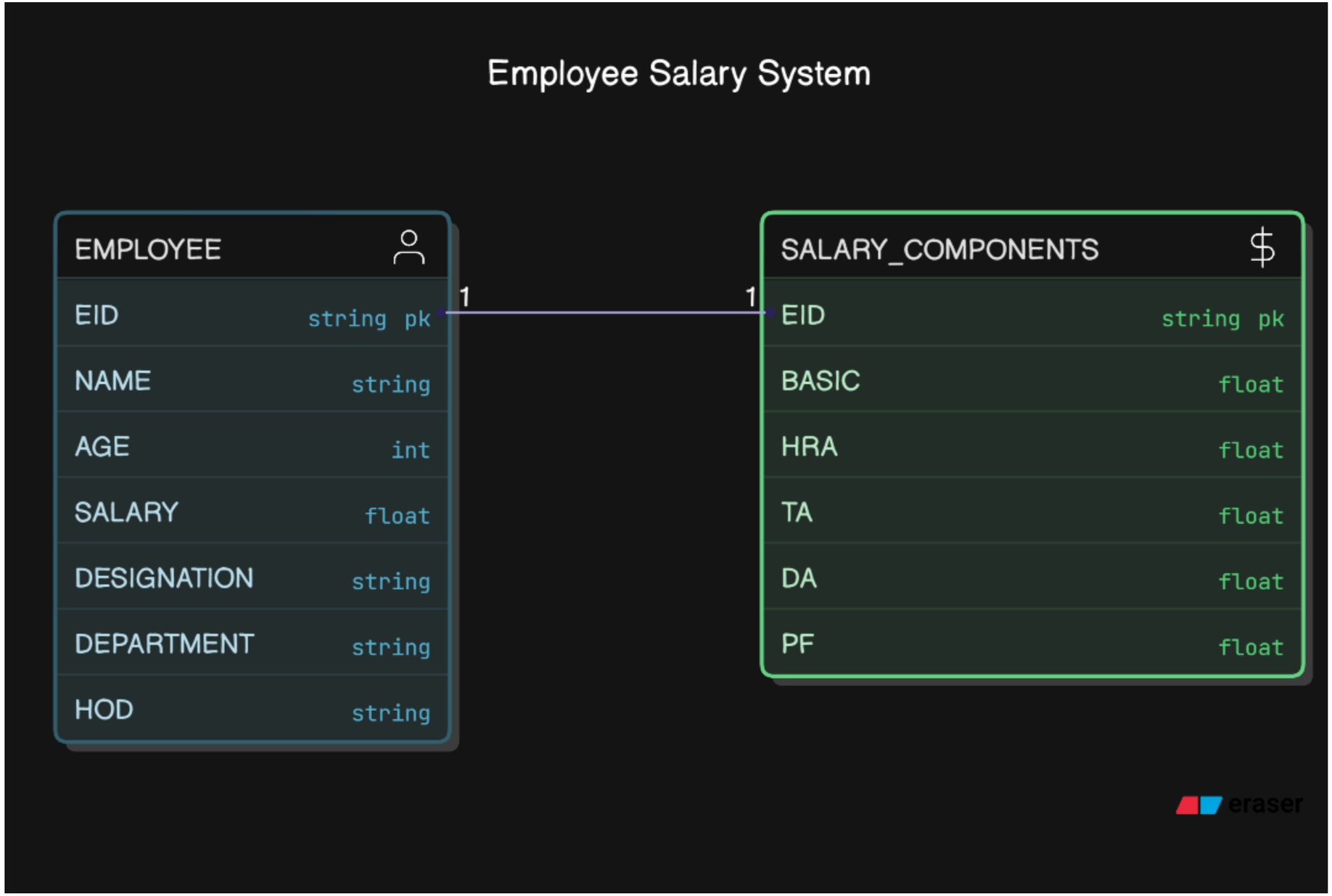
- The table Already satisfies this as all fields are atomic and cannot be divided more.



2NF

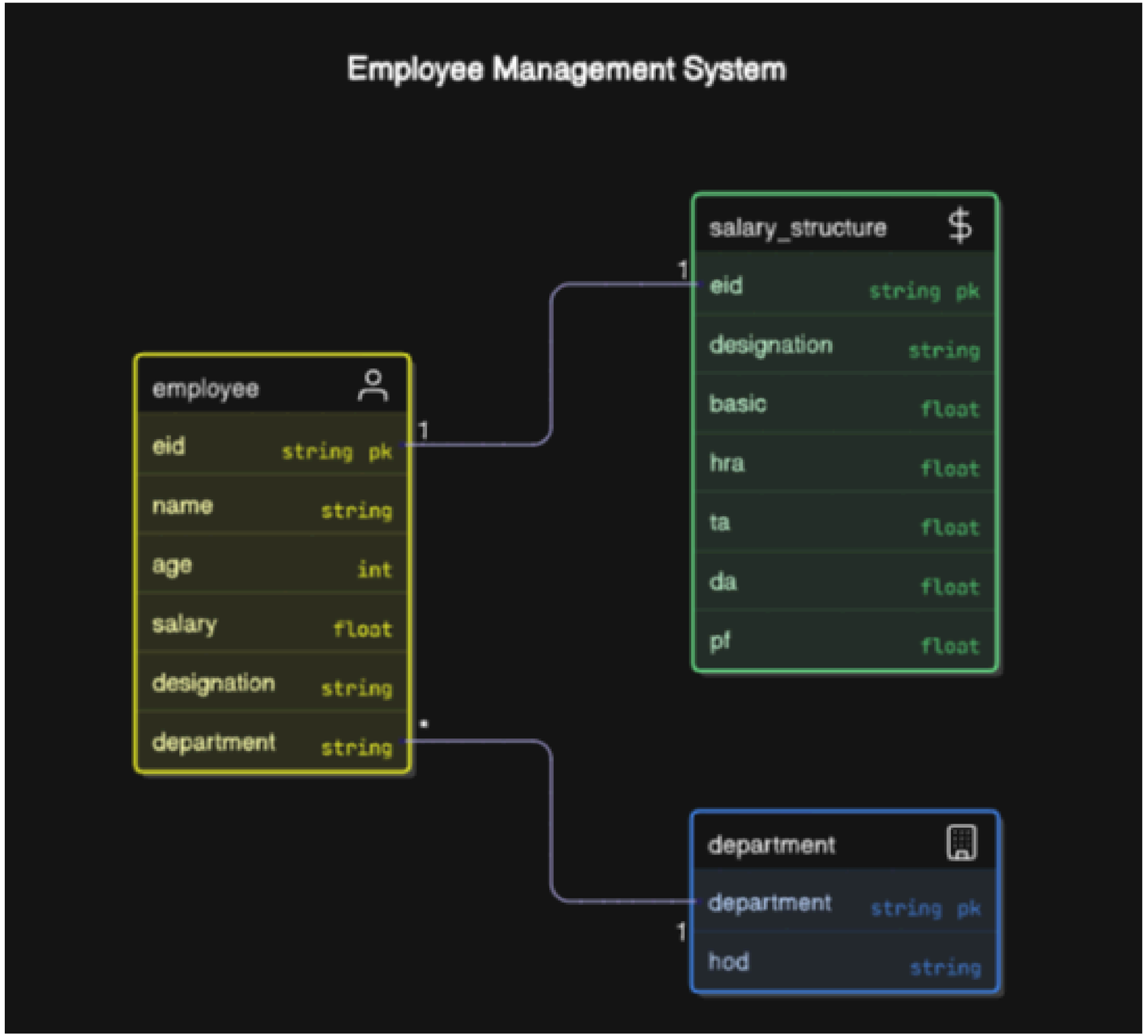
A table is in 2NF if it is in 1NF and all non-key attributes depend entirely on the primary key, not just a part of it (no partial dependencies).

- It's in 1NF already.
- EID is the primary key since each employee is individually identified by it.
- Since they only rely on EID, the properties NAME, AGE, SALARY, DESIGNATION, DEPARTMENT, and HOD are acceptable.
- However, the wage structure is dependent on the DESIGNATION rather than just EID and includes BASIC, HRA, TA, DA, and PF.


Employee Salary System
EMPLOYEE
EID string pk
NAME string
AGE int
SALARY float
DESIGNATION string
DEPARTMENT string
HOD string
1
1
SALARY_COMPONENTS
EID string pk
BASIC float
HRA float
TA float
DA float
PF float
eraser

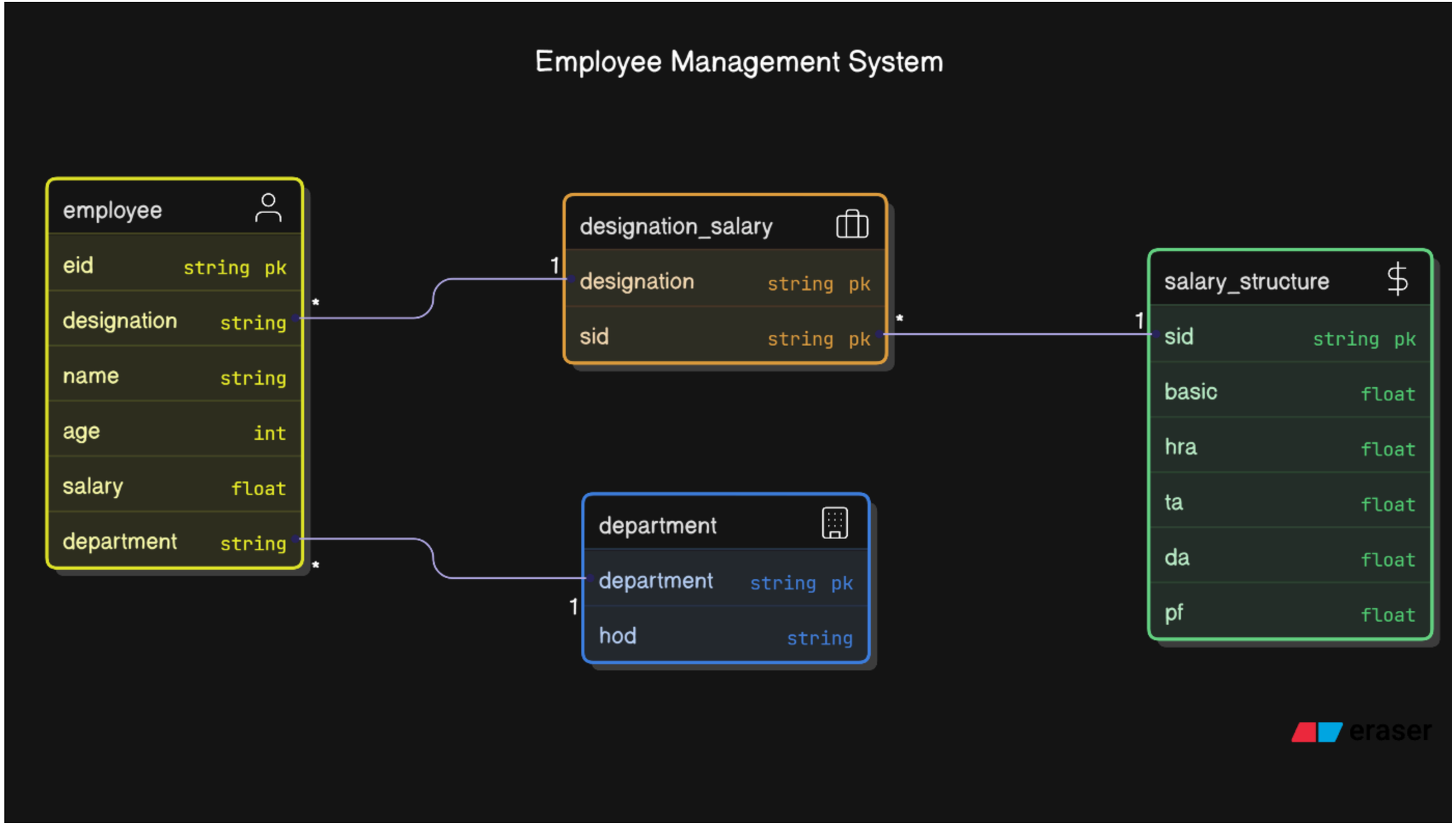
3NF

- A table is in 3NF there are no transitive dependencies (i.e., no non-key attribute depends on another non-key attribute).
- It's in 2NF already.
- There is a transitive dependency (EID → DEPARTMENT → HOD) as the DEPARTMENT decides HOD.
- Remove transitive dependencies:
- Designation Details: Create a DESIGNATION table to store salary components.
- Department Details: Create a DEPARTMENT table to store department-specific data like HOD.


Employee Management System
salary_structure
eid string pk
designation string
basic float
hra float
ta float
da float
pf float
employee
eid string pk
name string
age int
salary float
designation string
department string
department
department string pk
hod string
1
1
1
*

BCNF

- A table is in BCNF if it is in 3NF and every determinant (attribute that functionally determines another attribute) is a superkey.
- It's in 3NF already.
- DESIGNATION was not a superkey, so a new unique identifier (SID) was introduced.
- To maintain the relationship between DESIGNATION and salary structure, mapping each DESIGNATION to a unique SALARY_STRUCTURE (SID).


Employee Management System
employee
eid string pk
designation string
name string
age int
salary float
department string
designation_salary
designation string pk
sid string pk
salary_structure
sid string pk
basic float
hra float
ta float
da float
pf float
department
department string pk
hod string
1
*
1
*
1
*
1
eraser

What did this achieve?

- Eliminated Redundancy:
  - Removed repeated salary components for employees with the same designation.
  - Stored HOD details separately, avoiding duplication across employees.

- Removed Anomalies:
  - Insertion Anomaly: Allowed adding new designations without needing an employee entry.
  - Deletion Anomaly: Prevented loss of salary structure when an employee is removed.
  - Update Anomaly: Ensured changes in salary structure or HOD details do not require multiple updates.

- Improved Data Integrity:
  - Enforced correct relationships between employees, departments, and salary structures.
  - Ensured every determinant is a superkey, maintaining consistency.
- Optimized Storage & Query Performance:
  - Reduced data duplication, improving database efficiency.

Thank you