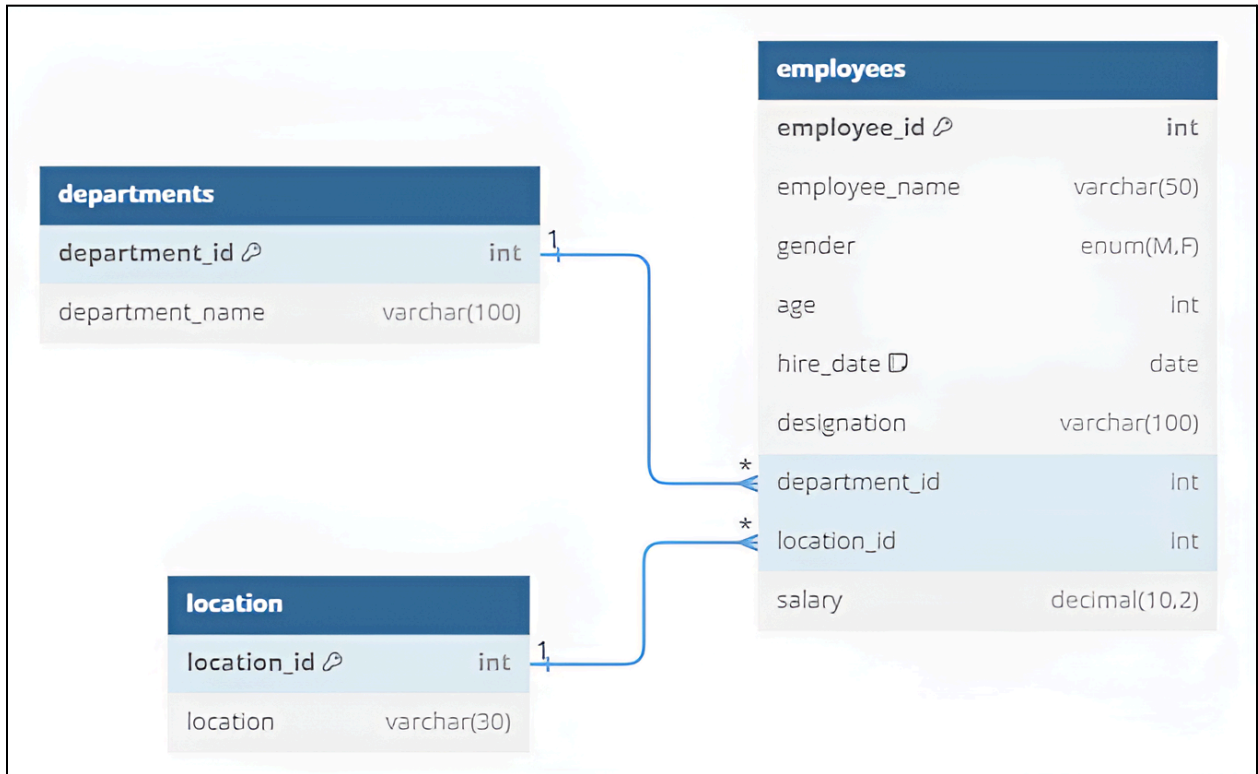# MySQL Assignment 1 – DDL Commands & Constraints

Employee Database schema:

DDL Commands :

1. **Table Creation (CREATE):** Write the SQL statements to create a database named "employee" and the following tables based on the provided schema:

   ⊙ Departments

   ⊙ Location

   ⊙ Employees

2. **Table Alteration (ALTER):** Consider the following scenarios and write the SQL statements to alter the structure of the tables accordingly:

   ⊙ Add a new column named "email" to the Employees table to store employee email addresses.

   ⊙ Modify the data type of the "designation" column in the Employees table to support a wider range of values.

   ⊙ Drop the "age" column from the Employees table.

   ⊙ Rename the "hire_date" column to "date_of_joining".

3. **Table Renaming (RENAME):** Rewrite the SQL statements to rename the following tables:

   ⊙ Rename the "Departments" table to "Departments_Info".

   ⊙ Rename the "Location" table to "Locations".

4. **Table Truncation (TRUNCATE):** Write an SQL statement to truncate the Employees table.

5. **Database & Table Dropping (DROP):** Write the SQL statements to drop the Employees table and then the "employee" database.

Constraints :

1. **Database Recreation:**

   ⊙ Drop the 'employee' database if it exists and recreate it using the provided schema, ensuring that all tables are created with the appropriate constraints as instructed.

2. **Departments Table:**

   ⊙ Ensure that the "department_id" uniquely identifies each department.

   ⊙ Set up constraints on the "department_name" to avoid duplicate and null entries.

3. **Location Table:**

   ⊙ Establish a mechanism to automatically generate unique identifiers for each location, ensuring that they are incremented sequentially.

   ⊙ Implement constraints to prevent the insertion of null and duplicate locations.

4. **Employees Table:**

   ⊙ Guarantee that each employee has a distinct identifier.

   ⊙ Create a restriction to ensure that the employee's name is always provided.

   ⊙ Limit the acceptable values for the "gender" field to only 'M' or 'F'.

   ⊙ Enforce a condition to ensure that the employee's age is 18 or above.

   ⊙ Automatically assign the current date to the "hire_date" field if not specified.

   ⊙ Establish links between the "department_id" and "location_id" fields in the "employees" table and their respective tables.