1. Employees Table

This table stores individual employee details, linking them to their respective departments and roles using foreign keys.

- **Primary Key (`id`)**: Ensures each employee record is unique.
- **Columns**:
  - `name`: Captures the name of the employee.
  - `email`: Stores a unique email for each employee to avoid duplication.
  - `department_id`: References the `departments` table to associate employees with their department.
  - `role_id`: References the `roles` table to associate employees with their assigned role.

2. Departments Table

This table defines the departments within the organization.

- **Primary Key (`id`)**: Uniquely identifies each department.
- **Columns**:
  - `name`: Provides the name of the department, ensuring uniqueness to avoid ambiguity.
  - `location`: (Optional) Stores the physical location of the department.

3. Roles Table

This table defines the various roles employees can have in the organization.

- **Primary Key (`id`)**: Uniquely identifies each role.
- **Columns**:
  - `name`: Defines the role name, such as Manager, Developer, etc.
  - `description`: (Optional) Provides a detailed explanation of the role's responsibilities.

4. Relationships

- **Employees and Departments**:
  - Each employee belongs to one department (`department_id` foreign key).
  - A department can have multiple employees, establishing a one-to-many relationship.
- **Employees and Roles**:
  - Each employee has one role (`role_id` foreign key).
  - A role can be assigned to multiple employees, creating a one-to-many relationship.

5. Data Types and Constraints

- `INT`: Used for IDs, ensuring efficient indexing and auto-increment functionality for unique identification.
- `VARCHAR`: Used for textual data like names and emails with length constraints to save storage space.

- `FOREIGN KEY`: Ensures data integrity by enforcing relationships between tables.
- `NOT NULL`: Applied to mandatory fields to prevent incomplete records.
- `UNIQUE`: Ensures values like emails and department names are not duplicated.

Benefits

- **Normalization**: Reduces redundancy by separating data into logical tables.
- **Flexibility**: Simplifies adding or modifying roles and departments without altering employee records.
- **Data Integrity**: Enforces valid relationships and prevents orphan records using foreign key constraints.