This schema includes:

An employees table with all required fields

A self-referencing manager\_id foreign key for the reporting structure

An email field for Gravatar integration

Timestamps for auditing

A hierarchy\_view for easier querying of the organizational structure

Software Architecture:

For this application, I recommend using a microservices architecture. This approach offers several benefits:

Scalability: Each service can be scaled independently.

Flexibility: Services can be developed, deployed, and maintained separately.

Resilience: If one service fails, it doesn't bring down the entire system.

Here's a high-level overview of the architecture:  
Design Patterns:

For this system, several design patterns would be beneficial:

a. Repository Pattern:

Use: For data access logic

Benefit: Separates the data access logic from business logic, making the code more maintainable and testable.

b. Factory Pattern:

Use: For creating complex objects, such as different types of employees or reports

Benefit: Provides a flexible way to create objects without exposing the creation logic to the client.

c. Observer Pattern:

Use: For updating the hierarchy view when employee data changes

Benefit: Allows for loose coupling between objects, making the system more flexible and easier to maintain.

d. Strategy Pattern:

Use: For implementing different sorting and filtering strategies in the reporting view

Benefit: Allows the algorithm to vary independently from clients that use it.

e. Decorator Pattern:

Use: For adding Gravatar functionality to employee profiles

Benefit: Allows for dynamically adding new behaviors to objects without altering their structure.

f. CQRS (Command Query Responsibility Segregation):

Use: Separating read and write operations for the employee data

Benefit: Can improve performance, scalability, and security by allowing the read and write models to scale independently.

CEO (top-level manager)

Three department managers (HR, Finance, IT) reporting to the CEO

Several employees under each department manager

An intern reporting to a senior developer