1. Explain the need and benefit of ORM

ORM (Object-Relational Mapping) is a programming technique that allows developers to interact with a relational database using the object-oriented paradigm of their programming language. It acts as a bridge between objects in code and tables in a database.

Need of ORM

- **Simplifies Database Interaction**

  ORM allows developers to use their programming language's syntax to interact with the database, eliminating the need to write complex SQL queries manually.

- **Object-Oriented Integration**

  It integrates seamlessly with object-oriented programming languages like Java, Python, or C#, allowing developers to work with data as objects.

- **Reduces Boilerplate Code**

  Common operations such as insert, update, delete, and fetch can be handled with minimal code, reducing redundancy and improving code readability.

- **Database Abstraction**

  ORM abstracts the underlying database, allowing applications to switch databases (e.g., MySQL to PostgreSQL) with minimal configuration changes.

- **Transaction Management**

  Most ORM frameworks offer built-in transaction support, enabling automatic handling of commit and rollback operations.

- **Security**

  ORM frameworks often use parameterized queries internally, helping prevent SQL injection attacks.

Benefits of Using ORM

- **Faster Development** : Reduces development time by minimizing SQL code and automating common tasks.
- **Maintainability** : Centralized access to database logic improves consistency and makes code easier to maintain
- **Code Reusability** : Entities and repositories can be reused across multiple services or components.
- **Portability** : Makes it easier to migrate the application to a different database system.
- **Improved Testing** : Simplifies testing by supporting in-memory databases and mock data handling.

Drawbacks of ORM

- **Performance Overhead** : Automatically generated queries may be less efficient than handwritten SQL.
- **Less Control** : Fine-tuning and optimizing complex queries can be difficult.
- **Learning Curve** : Requires understanding of ORM concepts, annotations, and configurations.