AEROASPIRE INTERN ASSESSMENT-02
KARTHIK Tl

1. **What is ORM, what are its advantages & disadvantages?**

   Object-Relational Mapping is a programming technique that acts as a bridge between object-oriented programming languages and relational databases .ORM lets developers interact with databases using familiar objects and class structures instead of hand-writing SQL. The ORM framework translates object code to SQL and back, mapping things like classes to tables and object attributes to columns.

   Advantages

   - Productivity: Developers can write less code for CRUD operations, as boilerplate SQL is handled by the ORM layer.
   - Abstraction: No need to write SQL for every operation. Code becomes more database-agnostic and easier to maintain.
   - Less Error-Prone: Stronger typing and reduced manual SQL means fewer SQL syntax/runtime mistakes.
   - Portability: ORM can support multiple types of databases, making it easier to switch databases by changing configuration instead of code.

   Disadvantages

   - Performance Overhead: The abstraction layer may generate suboptimal queries that can hurt performance compared to highly-tuned raw SQL.
   - Complex Query Limitations: For advanced queries using joins or custom logic, ORM APIs can become awkward or restrictive. You might need to drop down to direct SQL.
   - Learning Curve: Developers must learn both their ORM tool's conventions and, sometimes, its query language/syntax.

2. **How does parameterized query prevent SQL injection?**
   A parameterized query prevents SQL injection by separating SQL code from user data. Instead of inserting user input directly into the query string, it uses placeholders and sends the input separately. This way, the database treats user input strictly as data, not executable code, so injected malicious commands can't run. For example, an attacker’s input like '; DROP TABLE Users; -- is treated as a harmless string, not a command.

   The database compiles the SQL structure first, then binds the parameters, ensuring user input does not alter the query’s intent. This is a secure best practice to avoid SQL injection vulnerabilities.

AEROASPIRE INTERN ASSESSMENT-02
KARTHIK Tl

3. What is the flow from request → ORM / SQL → DB → return result → commit / rollback?

1. Request: The user sends a request to the application, triggering an action that requires a database interaction—like creating, updating, or retrieving data.
2. ORM / SQL: The application uses an ORM method or a direct SQL statement. If using ORM, it translates code into SQL behind the scenes. The query is prepared, often with parameter binding for safety.
3. DB: This SQL query is sent to the database server, which executes the statement.
4. Return result: The database returns the result—maybe a record, a list, or an acknowledgement to the application, and the ORM translates it back into objects if needed.
5. Commit / Rollback:
    - If all actions succeed, the transaction is committed, making all changes permanent.
    - If something fails (like a DB error or validation failure), the transaction is rolled back, undoing any parts that didn't fully succeed to keep the data consistent.