1. 
2. The following diagram shows the basic flow at the time of accessing the data from the database table using **Spring Data JPA**.
3. There are 6 steps in the flow.
4. A picture containing text, diagram, font, screenshot

   Description automatically generated
5. Let’s say that we want to save an entity into DB using Spring Data JPA.
6. **Step 01**: We will inject Spring Data JpaRepository into Service layer and we will call save method.  
   A picture containing text, screenshot, font, line

   Description automatically generated  
   ‘
7. **Step 02**: JpaRepository is interface, but we need implementation whose real method will be called like **save()**. So, we have two implementations:
   1. CustomRepositoryImpl.java and
   2. SimpleJpaRepository.java (Consider that we are using this class)
8. **Step 03**:
   1. Spring Data JPA calls internally the methods of JPA Specification.
   2. So, it means that Spring Data JPA is just a wrapper around the JPA to reduce the boilerplate code.
9. **Step 04**:
   1. But JPA is a just standard API, but we need its implementation.
   2. So, Spring Data JPA uses internally Hibernate by default as **JPA Provider**.
   3. To be very precise, JPA calls **Hibernate Core API**.
   4. Then hibernate will create SQL Statements.
10. **Step 05**:
    1. Then hibernate will calls JDBC methods.
11. **Step 06**:
    1. Eventually, the SQL statements are executed by JDBC API.
12. **NOTE**: Each step will pass on the parameters to the successive step.