@Temporall in Hibernate:

In Hibernate, storing Date and Time fields in the database requires explicit specification of how these fields should be interpreted and mapped. The @Temporal annotation is used to define the **temporal precision** of the Date or Calendar fields in a Java entity class.

This is necessary because the Date type in Java can represent both **date** and **time**, but databases usually differentiate between these as separate data types (DATE, TIME, TIMESTAMP). Without the @Temporal annotation, Hibernate cannot determine how to map the field correctly (default TIMESTAMP).

* @Temporal annotation comes with three TemporalType options:

1. **TemporalType.DATE:**
   * Maps the field to a(column type **DATE**) type in the database
   * Only the date part (year, month, day) will be stored, ignoring the time part.
   * Example: 2025-01-06.
2. **TemporalType.TIME:**
   * Maps the field to a(column type **TIME)** type in the database.
   * Only the time part (hour, minute, second) will be stored, ignoring the date part.
   * Example: 12:45:00.
3. **TemporalType.TIMESTAMP:**
   * Maps the field to a (column type **TIMESTAMP**)type in the database.
   * Stores both the **date** and **time** parts.
   * Example: 2025-01-06 12:45:00.