

Soluciones a error SQL 1265 y configuración ENUM en Spring Boot + MySQL

1■■■ Problema principal:

Spring Boot con JPA y MySQL puede crear columnas ENUM en la base de datos incluso cuando se usa `@Enumerated(EnumType.STRING)`. Esto causa errores como SQL Error 1265 (Data truncated) cuando se agregan nuevos valores al enum Java, ya que MySQL no actualiza automáticamente la definición de ENUM.

2■■■ Causas comunes:

- 1 El dialecto de Hibernate para MySQL genera ENUM nativos en lugar de VARCHAR.
- 2 Hibernate no actualiza los tipos ENUM cuando se agregan nuevos valores.
- 3 El modo de `hibernate.ddl-auto=update` no altera ENUMs existentes.
- 4 La base de datos rechaza valores no incluidos en la definición del ENUM.

3■■■ Soluciones posibles:

- 1 ■ ****Definir manualmente el tipo de columna como VARCHAR:****
`@Enumerated(EnumType.STRING) @Column(nullable = false, columnDefinition = 'varchar(50)')`
`private TypeAccount typeAccount;`
- 2 ■ ****Modificar el ENUM en la base de datos manualmente:**** `ALTER TABLE account MODIFY COLUMN type_account ENUM('SAVINGS','CHECKING','INVESTMENT') NOT NULL;`
- 3 ■ ****Usar VARCHAR para evitar problemas futuros:**** `ALTER TABLE account MODIFY COLUMN type_account VARCHAR(50);`
- 4 ■ ****Configurar Hibernate para no recrear ENUM:**** `spring.jpa.hibernate.ddl-auto=update`
`spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect`
- 5 ■ ****Usar scripts de migración (Flyway o Liquibase)**** para mantener sincronizados los valores de ENUM y evitar errores 1265.

4■■■ Recomendación final:

Para aplicaciones que cambian con frecuencia los valores de sus ENUM, lo más estable es usar columnas VARCHAR en lugar de ENUM SQL. Esto evita errores de inserción, facilita migraciones y mejora la compatibilidad con Hibernate.

5■■■ Comportamiento de ddl-auto:

• `create` → Borra y recrea todas las tablas (incluye los nuevos valores ENUM) • `update` → No modifica ENUM existentes • `validate` → Solo valida esquema • `none` → No hace cambios en la base de datos