# Analysis of Solutions for PostgreSQL v15.6 Upgrade/RHEL Migration Problem

## Problem Summary

The following error occurs due to a type mismatch between a PostgreSQL user-defined enum (`eccportal.usertype`) and an incoming value of type `character varying`. This is common in PostgreSQL when strict type enforcement clashes with Java Hibernate’s default handling of enums.

ERROR: column "type" is of type eccportal.usertype but expression is of type character varying  
Hint: You will need to rewrite or cast the expression

## Evaluating the Proposed Solutions

### Stack Overflow Solution

\*\*Proposed Fix:\*\* spring.datasource.hikari.data-source-properties.stringtype=unspecified

\*\*Pros:\*\* Quick and simple fix; Allows PostgreSQL to infer types.

\*\*Cons:\*\* May mask underlying issues; Less explicit.

\*\*Verdict:\*\* Potentially effective but risky.

### Claude Solution

\*\*Proposed Fix:\*\* Same as Stack Overflow + Update PostgreSQL JDBC driver

\*\*Pros:\*\* Aligns well with migration scenarios; Works well with Hibernate + PostgreSQL.

\*\*Cons:\*\* Workaround rather than structured fix; Can cause type inference issues.

\*\*Verdict:\*\* Good for quick fixes but not for long-term use.

### Perplexity Deep Research

\*\*Proposed Fix:\*\* 1. Custom Hibernate `UserType` (Hypersistence Utils)  
2. Explicit Casting via `@ColumnTransformer`  
3. JDBC `stringtype=unspecified`

\*\*Pros:\*\* Best structure (Hypersistence Utils); Safe conversion (`@ColumnTransformer`).

\*\*Cons:\*\* Requires dependency (`Hypersistence Utils`); Minor entity modifications.

\*\*Verdict:\*\* Highly recommended for production.

### Google AI Studio Solution

\*\*Proposed Fix:\*\* Investigate `eccportal.usertype` structure; Ensure proper ORM mapping; Consider explicit casting or schema changes.

\*\*Pros:\*\* Deep investigation before applying a fix; Provides structured ORM-based solutions.

\*\*Cons:\*\* Less specific on immediate fixes; May require database schema modifications.

\*\*Verdict:\*\* Good for debugging, not an immediate fix.

## Final Recommendations

|  |  |  |  |
| --- | --- | --- | --- |
| Solution | Effectiveness | Complexity | Best Use Case |
| Hypersistence Utils (`PostgreSQLEnumType`) | ✅✅✅ (Best) | 🟠 (Moderate) | Production, maintainability-focused migration |
| @ColumnTransformer (Explicit Casting) | ✅✅ | 🟠 (Moderate) | Small projects with minor entity modifications |
| JDBC `stringtype=unspecified` | ✅ (Temporary) | 🟢 (Easy) | Quick fix, but not recommended for long-term |
| Google AI Studio Investigative Approach | ✅✅ | 🔴 (High) | Structured debugging and database schema understanding |

## Conclusion

If long-term stability and maintainability are priorities, use Hypersistence Utils or `@ColumnTransformer`. If a quick fix is needed, `stringtype=unspecified` can work but should not be relied on for production. If the database schema is flexible, modifying the column type is another potential solution.