# Camunda 8 Zeebe Custom RDBMS Exporter Implementation Summary

Date: June 26, 2025

## 1. Overview

This document summarizes the development and integration of a custom RDBMS exporter for Camunda 8 Zeebe, specifically to export process and user task records to an Oracle database. The exporter listens to Zeebe engine events and maps record values into structured database tables.

## 2. Steps of Implementation

* 1. Created a Maven-based Java project named `DBExporter`.
* 2. Added required dependencies: `zeebe-exporter-api`, `zeebe-protocol-jackson`, and Oracle JDBC driver (`ojdbc17.jar`).
* 3. Implemented the `Exporter` interface and its `export(Record<?> record)` method.
* 4. Added `log4j2.xml` configuration inside Zeebe to capture logs from the custom exporter.
* 5. Built the exporter JAR and placed it inside Zeebe broker lib directory.
* 6. Updated Zeebe `application.yaml` to configure the custom exporter class with JDBC connection string to Oracle DB.
* 7. Mapped the values from `UserTaskRecordValue` and `ProcessInstanceRecordValue` into insert/update SQL queries.
* 8. Logged the incoming record types and handled record-specific mapping using instanceof checks.
* 9. For date fields like `dueDate`, parsed string format (e.g., '2025-06-19T21:07:00.000-0500') using correct DateTimeFormatter.
* 10. Ran process instances in Zeebe to trigger exporter logic and validated database insertions.

## 3. Record Value Knowledge

Camunda Zeebe records received in the exporter are of type `Record<?>`. Using `record.getValue()` we typecast to specific value types like `UserTaskRecordValue` or `ProcessInstanceRecordValue`. These record values contain metadata and business fields like task key, name, process key, due date, state, and more.

## 4. Mapping Logic

Using `PreparedStatement`, we extract and set values like:  
- Key: `record.getKey()`  
- Element ID: `userTask.getElementId()`  
- Completion Timestamp: parsed from `record.getTimestamp()` if state is `COMPLETED`  
- Due Date: parsed using `DateTimeFormatter.ofPattern("yyyy-MM-dd'T'HH:mm:ss.SSSZ")`  
- Process/Task metadata and custom headers

## 5. Data Loss and Risk Considerations

The exporter runs inside the Zeebe broker process. If the DB connection fails or an exception is thrown during export, the following issues may arise:

* 1. Zeebe engine continues processing the workflow without retrying the failed export.
* 2. Exporter does not automatically reprocess old events once restarted, unless explicitly reconfigured.
* 3. Any records that failed during insertion will not be retried unless custom logic is added.
* 4. Proper exception handling and alerting must be implemented to ensure data integrity.

## 6. Proof of Implementation

Logs from `zeebe.log` show that the exporter was initialized and received events such as `UserTaskRecordValue`. Print statements and logger outputs confirm the data structure and SQL insertions were triggered on task creation.

Example Log Snippet:

[Exporter] record value: {"id":"2251799813689678", "name":"User Task", "dueDate":"2025-06-19T21:07:00.000-0500"...}

## 7. Conclusion

The custom RDBMS exporter has been successfully implemented and integrated with Oracle DB. It listens to Zeebe events and stores relevant metadata into structured tables. Future improvements can include retry logic, better exception handling, and support for more record types.