LzRelational™

Many organizations that continue to depend on the mainframe for their enterprise applications use the IBM DB2® RDBMS for their mission-critical data. LzRelational provides support for relational schemas using PostgreSQL, an open-source RDBMS.

LzRelational

LzRelational provides support for relational schemas using PostgreSQL, an open-source RDBMS. LzRelational uses PostgreSQL user-defined types to provide compatibility with the data types and functions uniquely found in DB2 mainframes, enabling customer mainframe applications and data to be easily rehosted to the new environment.

Product Overview

LzRelational is a relational database implementation within the LzLabs Software Defined Mainframe® (SDM) based on the open-source relational database PostgreSQL (v11). PostgreSQL shares many concepts and features with DB2 for IBM z/OS.

All LzRelational services are built on standard PostgreSQL facilities and interfaces. LzRelational leverages user-defined data types, functions and operators for compatibility with DB2 for z/OS while adding support for the DB2 SQL dialect. No modifications were made to the core PostgreSQL solution.

Key Benefits

LzRelational provides customers with the following advantages:

- Low cost, functionally equivalent platform.
- Minimal risk when rehosting customer mainframe application as minimal changes to code and data are required.
- Numerous modernization options on enterprise class x86/Linux® or cloud infrastructures.
- Straightforward integration of analytics and Big Data platforms with mainframe data on top of the PostgreSQL environment.
- Significant use of open-source solutions to drive real business innovation and reduce IT technical debt.
- Manageable by modern IT personnel: required competences closely suited to current job market: new DBAs can be recruited with a PostgreSQL profile rather than IBM DB2 expertise.

Technical Benefits

- High data fidelity through preservation of original encoding format (EBCDIC/ASCII/UNICODE), CCSIDs, and the preservation of original SQL dialect.
- Support for JCL; no migration to a scripting language is required.

Features Overview

The following features are provided:

- LzRelational transforms the application's SQL statements (in DB2 for z/OS dialect) into PostgreSQL dialect as required:
  - Static SQL – DB2 for z/OS SQL will be transformed into PostgreSQL syntax during data migration or when an LzRelational release/ version update is performed.
  - Dynamic SQL – SQL statements will be transformed into PostgreSQL syntax during execution.
- Database operations are supported from a variety of environments (or sources), e.g., a COBOL, PL/I, or Assembler batch program, a CICS Online application, an IMS/TM application, or an ODBC/JDBC request.
- Interactive execution of SQL statements are supported via a management console (UI).
- Support for the execution of PostgreSQL on a separate Linux instance from the SDM.
- Single - and two-phase commit are supported, along with restart processing through LzOnline™.
- LzLabs-exclusive PostgreSQL data types enable DB2 compatibility.
- Essential utilities equivalent to those provided by DB2 are provided; see the LzRelational Utilities Guide for details.

LzRelational

Connections

LzRelational supports several different attach methods to mirror legacy system capabilities:

- Call Attach Facility (CAF).
- Distributed Relational Database Architecture (DRDA) – ODBC/JDBC.
- Rest Server (REST).
- RRS Attach Facility (RRSAF).
- Single Address Space (SASS) – LzOnline.
- MASS (LzOnline).

Dynamic SQL

- Executed through a batch program, LzOnline, SDM management console or DRDA (for remote application in Java, .Net, etc.)
- Dynamic SQL can be executed through the LzLabs REST API or via batch programs such as DSNTEP2.

Data Types

The following data types are supported:

- Numbers: SMALLINT, INTEGER, BIGINT, REAL, DOUBLE/FLOAT, DECIMAL(p, s), DECFLOAT (partial support).
- Strings: CHAR(n), VARCHAR(n).
- Large Objects: CLOB, BLOB (both 1 GiB Max).
- Date/Time: DATE, TIME, TIMESTAMP(p) WITHOUT TIME ZONE.
- ROWID

Note: These data types allow LzRelational to provide the proper sort order and comparison operations. This allows for similar manipulation of the application's date and time values.

Monitoring

- Current thread activity monitoring and display.
- DBMS-level statistics data collection and display.
- Thread-, Package-, and Statement-level accounting data collection that can be externalized as custom LzMetrics™ (SMF-style) records with on-demand conversion to z/OS DB2 SMF-compatible type 101 IFCID 003 and 239 records.
- Online historical thread activity reporting that supports drill-down to the SQL statement level.
- LzMetrics Export utility support for exporting SDM/LzRelational accounting data into Linux CSV files for post-processing outside the SDM.
- LzMetrics Aggregation utility support for generating summarized versions of SDM/LzRelational accounting data.

Contact Us

LinkedIn: LzLabs GmbH
Twitter: @LzLabsGmbH

info@lzlabs.com

LzLabs GmbH
Richtiarkade 16
CH-8304 Wallisellen
Switzerland
Tel: +41 44 515 9880

LzLabs
25 Templer Avenue
Farnborough
GU14 6FE
United Kingdom
Tel: +44 (0)1252 917232

lzlabs.com/products