

Preface

This manual describes the creation and use of updatable JSON views of relational data stored in Oracle Database. The view data has a JSON-relational **duality**: it's organized both relationally and hierarchically. The manual covers how to create, query, and update such views, which automatically entails updating the underlying relational data.

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Documents](#)

Oracle and other resources related to this developer's guide are listed.

- [Conventions](#)

- [Code Examples](#)

The code examples here are for illustration only, but in many cases you can copy, paste, and run parts of them in your environment. Unless called out explicitly, the examples do not depend on each other in any way. In particular, there is no implied sequencing among them.

Audience

JSON-Relational Duality Developer's Guide is intended mainly for two kinds of developers: (1) those building applications that directly use data in relational tables, but who also want to make some of that table data available in the form of JSON document collections, and (1) those building applications that directly use JSON documents whose content is based on relational data.

An understanding of both JavaScript Object Language (JSON) and some relational database concepts is helpful when using this manual. Many examples provided here are in Structured Query Language (SQL). A working knowledge of SQL is presumed.

Some familiarity with the [GraphQL](#) language and REST (REpresentational State Transfer) is also helpful. Examples of creating JSON-relational duality views are presented using SQL and, alternatively, a subset of GraphQL. Examples of updating and querying JSON documents that are supported by duality views are presented using SQL and, alternatively, REST requests.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

Oracle and other resources related to this developer's guide are listed.

- *Oracle Database JSON Developer's Guide*
- *Migrating From JSON To Duality in Oracle Database Utilities*
- Product page [Simple Oracle Document Access \(SODA\)](#) and book *Oracle Database Introduction to Simple Oracle Document Access (SODA)*
- Product page [Oracle Database API for MongoDB](#) and book *Oracle Database API for MongoDB*
- Product page [Oracle REST Data Services \(ORDS\)](#) and book *Oracle REST Data Services Developer's Guide*
- *Oracle Database SQL Language Reference*
- *Oracle Database PL/SQL Language Reference*
- *Oracle Database PL/SQL Packages and Types Reference*
- *Oracle Database Concepts*
- *Oracle Database Error Messages Reference*. Oracle Database error message documentation is available only as HTML. If you have access to only printed or PDF Oracle Database documentation, you can browse the error messages by range. Once you find the specific range, use the search (find) function of your Web browser to locate the specific message. When connected to the Internet, you can search for a specific error message using the error message search feature of the Oracle Database online documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at OTN Registration.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Code Examples

The code examples here are for illustration only, but in many cases you can copy, paste, and run parts of them in your environment. Unless called out explicitly, the examples do not depend on each other in any way. In particular, there is no implied sequencing among them.

- [Pretty Printing of JSON Data](#)
To promote readability, especially of lengthy or complex JSON data, output is sometimes shown pretty-printed (formatted) in code examples.
- [Reminder About Case Sensitivity](#)
JSON is case-sensitive. SQL is case-insensitive, but names in SQL code are implicitly uppercase.

Pretty Printing of JSON Data

To promote readability, especially of lengthy or complex JSON data, output is sometimes shown pretty-printed (formatted) in code examples.

Reminder About Case Sensitivity

JSON is case-sensitive. SQL is case-insensitive, but names in SQL code are implicitly uppercase.

When examining the examples in this book, keep in mind the following:

- SQL is case-insensitive, but names in SQL code are implicitly uppercase, unless you enclose them in double quotation marks ("").
- JSON is case-sensitive. You must refer to SQL names in JSON code using the correct case: uppercase SQL names must be written as uppercase.

For example, if you create a table named `my_table` in SQL without using double quotation marks, then you must refer to it in JSON code as `"MY_TABLE"`.