

# Oracle® Database

## JSON-Relational Duality Developer's Guide



23ai  
F57229-11  
April 2025

ORACLE®

Copyright © 2023, 2025, Oracle and/or its affiliates.

Primary Author: Drew Adams

Contributors: Oracle JSON development, product management, and quality assurance teams.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Audience	ix
Documentation Accessibility	ix
Diversity and Inclusion	x
Related Documents	x
Conventions	x
Code Examples	xi

## 1 Overview of JSON-Relational Duality Views

---

1.1 Table-Centric Use Case for JSON-Relational Duality	1-4
1.2 Document-Centric Use Case for JSON-Relational Duality	1-7
1.3 Map JSON Documents, Not Programming Objects	1-10
1.4 Duality-View Security: Simple, Centralized, Use-Case-Specific	1-12
1.5 Oracle Database: Converged, Multitenant, Backed By SQL	1-13

## 2 Introduction To Car-Racing Duality Views Example

---

2.1 Car-Racing Example, JSON Documents	2-2
2.2 Car-Racing Example, Entity Relationships	2-6
2.3 Car-Racing Example, Tables	2-8

## 3 Creating Duality Views

---

3.1 Creating Car-Racing Duality Views Using SQL	3-4
3.2 Creating Car-Racing Duality Views Using GraphQL	3-8
3.3 WHERE Clauses in Duality-View Tables	3-14

## 4 Updatable JSON-Relational Duality Views

---

4.1 Annotations (NO)UPDATE, (NO)INSERT, (NO)DELETE, To Allow/Disallow Updating Operations	4-2
4.2 Annotation (NO)CHECK, To Include/Exclude Fields for ETAG Calculation	4-3
4.3 Database Privileges Needed for Duality-View Updating Operations	4-5

## 5 Using JSON-Relational Duality Views

---

5.1	Inserting Documents/Data Into Duality Views	5-3
5.2	Deleting Documents/Data From Duality Views	5-10
5.3	Updating Documents/Data in Duality Views	5-13
5.3.1	Trigger Considerations When Using Duality Views	5-22
5.4	Using Optimistic Concurrency Control With Duality Views	5-23
5.4.1	Using Duality-View Transactions	5-32
5.5	Using the System Change Number (SCN) of a JSON Document	5-36
5.6	Optimization of Operations on Duality-View Documents	5-38
5.7	Obtaining Information About a Duality View	5-40

## 6 Document-Identifier Field for Duality Views

---

## 7 Generated Fields, Hidden Fields

---

## 8 Schema Flexibility with JSON Columns in Duality Views

---

8.1	Embedding Values from JSON Columns into Documents	8-2
8.2	Merging Fields from JSON Flex Columns into Documents	8-4
8.3	When To Use JSON-Type Columns for a Duality View	8-6
8.4	Flex Columns, Beyond the Basics	8-8

## 9 GraphQL Language Used for JSON-Relational Duality Views

---

9.1	Oracle GraphQL Directives for JSON-Relational Duality Views	9-4
9.1.1	Oracle GraphQL Directive @link	9-6

## Index

---

## List of Examples

---

2-1	A Team Document	2-3
2-2	A Driver Document	2-3
2-3	A Car-Race Document	2-4
2-4	Creating the Car-Racing Tables	2-10
3-1	Creating Duality View TEAM_DV Using SQL	3-5
3-2	Creating Duality View DRIVER_DV, With Nested Team Information Using SQL	3-5
3-3	Creating Duality View DRIVER_DV, With Unnested Team Information Using SQL	3-6
3-4	Creating Duality View RACE_DV, With Nested Driver Information Using SQL	3-7
3-5	Creating Duality View RACE_DV, With Unnested Driver Information Using SQL	3-7
3-6	Creating Duality View TEAM_DV Using GraphQL	3-12
3-7	Creating Duality View DRIVER_DV Using GraphQL	3-12
3-8	Creating Duality View RACE_DV Using GraphQL	3-13
3-9	WHERE Clause Use in Duality View Definition (SQL)	3-15
3-10	WHERE Clause Use in Duality View Definition (GraphQL)	3-16
5-1	Inserting JSON Documents into Duality Views, Providing Document-Identifier Fields — Using SQL	5-5
5-2	Inserting JSON Documents into Duality Views, Providing Document-Identifier Fields — Using REST	5-6
5-3	Inserting JSON Data into Tables	5-8
5-4	Inserting a JSON Document into a Duality View Without Providing Document-Identifier Fields — Using SQL	5-8
5-5	Inserting a JSON Document into a Duality View Without Providing Document-Identifier Fields — Using REST	5-9
5-6	Deleting a JSON Document from Duality View RACE_DV — Using SQL	5-11
5-7	Deleting a JSON Document from Duality View RACE_DV — Using REST	5-12
5-8	Updating an Entire JSON Document in a Duality View — Using SQL	5-15
5-9	Updating an Entire JSON Document in a Duality View — Using REST	5-16
5-10	Updating Part of a JSON Document in a Duality View	5-17
5-11	Updating Interrelated JSON Documents — Using SQL	5-18
5-12	Updating Interrelated JSON Documents — Using REST	5-19
5-13	Attempting a Disallowed Updating Operation Raises an Error — Using SQL	5-20
5-14	Attempting a Disallowed Updating Operation Raises an Error — Using REST	5-21
5-15	Obtain the Current ETAG Value for a Race Document From Field etag — Using SQL	5-28
5-16	Obtain the Current ETAG Value for a Race Document From Field etag — Using REST	5-28
5-17	Using Function SYS_ROW_ETAG To Optimistically Control Concurrent Table Updates	5-29
5-18	Locking Duality-View Documents For Update	5-34
5-19	Using a Duality-View Transaction To Optimistically Update Two Documents Concurrently	5-34

---

5-20	Obtain the SCN Recorded When a Document Was Fetched	5-36
5-21	Retrieve a Race Document As Of the Moment Another Race Document Was Retrieved	5-37
5-22	Using DBMS_JSON_SCHEMA.DESCRIBE To Show JSON Schemas Describing Duality Views	5-42
6-1	Document Identifier Field _id With Primary-Key Column Value	6-1
6-2	Document Identifier Field _id With Object Value	6-2
7-1	Fields Generated Using a SQL Query and a SQL Expression	7-2
7-2	Field Generated Using a SQL/JSON Path Expression	7-3
7-3	Fields Generated Using Hidden Fields	7-4
9-1	Creating Duality View DRIVER_DV1, With Nested Driver Information	9-5
9-2	Creating Table TEAM_W_LEAD With LEAD_DRIVER Column	9-7
9-3	Creating Duality View TEAM_DV2 With LEAD_DRIVER, Showing GraphQL Directive @link	9-8
9-4	Creating Duality View DRIVER_DV2, Showing GraphQL Directive @link	9-9
9-5	Creating Table DRIVER_W_MGR With Column MANAGER_ID	9-10
9-6	Creating Duality View TEAM_DV3 (Drivers with Managers)	9-11
9-7	Creating Duality View DRIVER_DV3 (Drivers with Managers)	9-12
9-8	Creating Duality View DRIVER_MANAGER_DV	9-13

## List of Figures

---

2-1	Car-Racing Example, Directed Entity-Relationship Diagram (1)	2-7
2-2	Car-Racing Example, Directed Entity-Relationship Diagram (2)	2-10
3-1	Car-Racing Example, Table-Dependency Graph	3-9
5-1	Optimistic Concurrency Control Process	5-26
9-1	Car-Racing Example With Team Leader, Table-Dependency Graph	9-7
9-2	Car-Racing Example With Driver Self-Reference, Table-Dependency Graph	9-10

## List of Tables

---

9-1 Scalar Types: Oracle JSON, GraphQL, and SQL

9-2



# 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
<b>boldface</b>	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"`.