

Index

Symbols

`_id` field, document-identifier, [6-1](#)
`_id` field, row-identifier, [6-1](#)
`_metadata` field, for document handling, [3-1](#), [5-23](#)
`_nameConflicts` field, for flex-column conflicts, [8-8](#)
`@delete` annotation (GraphQL), [3-8](#)
`@flex` annotation, [8-8](#)
`@flex` GraphQL directive, [9-4](#)
`@generated` GraphQL directive, [9-4](#)
`@hidden` GraphQL directive, [9-4](#)
`@insert` annotation (GraphQL), [3-8](#)
`@link` GraphQL directive, [3-8](#), [9-4](#), [9-6](#)
`@nest` GraphQL directive, [9-4](#)
`@unnest` GraphQL directive, [3-8](#), [9-4](#)
`@update` annotation (GraphQL), [3-8](#)

Numerics

1:1 entity relationships, [2-6](#)
1:N entity relationships, [2-6](#)

A

`ALL_JSON_DUALITY_VIEW_TAB_COLS` view, [5-40](#)
`ALL_JSON_DUALITY_VIEW_TABS` view, [5-40](#)
`ALL_JSON_DUALITY_VIEWS` view, [5-40](#)
annotations
 `@delete` (GraphQL), [3-8](#)
 `@flex` (GraphQL), [8-8](#)
 `@insert` (GraphQL), [3-8](#)
 `@update` (GraphQL), [3-8](#)
 AS FLEX (SQL), [8-8](#)
 ETAG, [4-2](#), [8-6](#), [8-8](#)
 updatability, [4-2](#), [8-6](#), [8-8](#)
application migration to using duality views, [2-1](#)
AS FLEX annotation, [8-8](#)
`asof` field, system change number (SCN), [3-1](#)
 ensuring read consistency, [5-36](#)
associative table
 See mapping table

B

bracket, optional GraphQL syntax for duality view
 definition, [3-8](#)
bridge table
 See mapping table

C

car-racing example, [2-1](#)
 creating duality views with GraphQL, [3-8](#)
 creating duality views with SQL, [3-4](#)
 creating tables, [2-8](#)
 duality views, [3-1](#)
 entity relationships, [2-6](#)
case-sensitivity
 JSON and SQL, [xi](#)
CHECK annotation (ETAG calculation), [4-3](#)
column types in tables underlying duality views, [2-8](#)
columns (hidden) for duality-view, ETAG and
 object ID, [3-1](#)
comment, GraphQL, [3-8](#)
complex or simple underlying data, [8-6](#)
composite primary and foreign keys, definition, [2-8](#)
concurrency, controlling, [5-23](#), [5-32](#)
content-based ETAG concurrency control,
 definition, [5-23](#)
converged database, definition, [1-13](#)
`CREATE JSON RELATIONAL DUALITY VIEW`,
 [3-1](#)
 WHERE clause, [3-14](#)
creating duality views, [3-1](#)
 over existing relational data, [1-4](#)

D

d-r-map entity, [2-8](#)
DATA JSON-type column for duality-view
 documents, [3-1](#)
DATA payload JSON-type column supported/
 generated by a duality view, [3-1](#), [5-38](#)
data types of columns in tables underlying duality
 views, [2-8](#)

DBA_JSON_DUALITY_VIEW_TAB_COLS view, [5-40](#)

DBA_JSON_DUALITY_VIEW_TABS view, [5-40](#)

DBA_JSON_DUALITY_VIEWS view, [5-40](#)

DBMS_JSON_DUALITY.begin_transaction
PL/SQL procedure, [5-32](#)

DBMS_JSON_DUALITY.commit_transaction
PL/SQL procedure, [5-32](#)

DBMS_JSON_DUALITY.register PL/SQL
procedure, [5-32](#)

DBMS_JSON_SCHEMA.describe PL/SQL
function, [1-7](#), [5-40](#)

DELETE annotation, [4-2](#)

deleting documents, [5-10](#)

describe PL/SQL function, package
DBMS_JSON_SCHEMA, [1-7](#), [5-40](#)

dictionary views, [1-7](#)

directives, GraphQL
See GraphQL directives

document
deleting, [5-10](#)
inserting, [5-3](#)
optimizing operations, [5-38](#)
querying, [5-38](#)
updating, [5-13](#)

document collection supported by a duality view,
definition, [1-1](#)

document key
definition, [3-1](#)

document migration to duality views, [2-1](#)

document-centric use of duality views, [1-7](#)

document-handling field, _metadata, [3-1](#), [5-23](#)

document-identifier field, _id, [6-1](#)

document-identifier field, car-racing example, [2-2](#)

document-relational mapping (DRM), definition,
[1-10](#)

document-version identifier (ETAG value), [3-1](#)

document-version identifier (ETAG)
controlling concurrency, [5-23](#)

document/table duality, definition, [1-1](#), [1-10](#)

documents supported by a duality view, definition,
[1-1](#)

documents, car-racing example, [2-2](#)

driver and race mapping table, [2-8](#)

driver document, [2-2](#)

driver duality view, [3-1](#)
creating with GraphQL, [3-8](#)
creating with SQL, [3-4](#)
JSON schema, [5-40](#)

driver entity, [2-6](#)

driver table, [2-8](#)

driver_race_map table, [2-8](#)

duality view, [1-1](#)
creation, [3-1](#)
definition, [1-1](#), [1-7](#)
JSON schema, [5-40](#)

duality view (*continued*)
motivation, [1-7](#)
overview, [1-1](#)
privileges needed for updating, [4-5](#)
rules for updating, [4-6](#)

duality view operations, [5-1](#)

duality views for car-racing example, [3-1](#)
creating with GraphQL, [3-8](#)
creating with SQL, [3-4](#)

duality, document/table, [1-10](#)
definition, [1-1](#)

E

embedding JSON-column values in documents,
[8-1](#), [8-2](#)

entity relationships, [2-6](#)

ETAG document-version identifier
controlling concurrency, [5-23](#)
not used for partial updates, [5-13](#)

etag field, version identifier, [3-1](#)
controlling concurrency, [5-23](#)

ETAG hash-value participation, defining, [4-3](#)

ETAG hidden duality-view column for ETAG value,
[3-1](#), [5-32](#)

ETAG table-row value, [5-23](#)

ETAG value, document-version identifier, [3-1](#)

evolution, schema, [8-8](#)

existing relational data as starting point, [1-4](#)

F

fields, [3-1](#)
_id, document-identifier, [6-1](#)
_id, row-identifier, [6-1](#)
_metadata, for document handling, [3-1](#), [5-23](#)
_nameConflicts for flex-column conflicts, [8-8](#)
asof, system change number (SCN), [3-1](#)
ensuring read consistency, [5-36](#)
etag, version identifier, [3-1](#)
controlling concurrency, [5-23](#)

flex column
definition, [8-4](#)
details, [8-8](#)

flex-column, field-naming conflicts, [8-8](#)

flexibility, schema, [8-2](#), [8-4](#), [8-6](#), [8-8](#)

foreign key, definition, [2-8](#)

Formula 1 example, [2-1](#)

function SYS_ROW_ETAG, optimistic
concurrency control, [5-23](#)

G

generated JSON fields, [7-1](#)

generation functions, SQL/JSON, [3-1](#)

GraphQL

- comment, [3-8](#)
- creating car-racing duality views, [3-8](#)
- creating duality views, [9-1](#)
- optional bracket syntax for duality view definition, [3-8](#)

GraphQL directives, [9-4](#)

- @flex, [9-4](#)
- @generated, [9-4](#)
- @hidden, [9-4](#)
- @link, [3-8](#), [9-4](#), [9-6](#)
- @nest, [9-4](#)
- @unnest, [3-8](#), [9-4](#)

H

hidden duality-view columns for ETAG and object ID, [3-1](#)

hidden JSON fields, [7-1](#)

I

- identifying column, definition, [2-8](#)
- identity column, definition, [6-1](#)
- indentifying column, definition, [2-2](#), [3-1](#)
- inline augmentation, [7-1](#)
- INSERT annotation, [4-2](#)
- inserting documents, [5-3](#)
- item methods, used to optimize operations, [5-38](#)

J

- JSON data type columns in duality-view tables, [1-1](#), [1-7](#), [8-1](#), [8-2](#), [8-6](#), [8-8](#)
- JSON documents, car-racing example, [2-2](#)
- JSON schema
 - use to validate JSON-column data, [1-7](#), [2-8](#), [8-2](#), [8-6](#), [8-8](#)
- JSON Schema, [1-7](#), [5-13](#), [8-2](#), [8-6](#), [8-8](#)
 - description of duality view, [5-40](#)
- JSON schema, use to validate JSON-column data, [5-13](#)
- JSON type modifiers, [8-2](#)
 - OBJECT, [8-8](#)
- JSON_SCHEMA column, dictionary views for duality views, [1-7](#), [5-40](#)
- json_transform SQL function, [5-13](#)
- json_value RETURNING clause, used to optimize operations, [5-38](#)
- JSON-relational duality view
 - definition, [1-1](#), [1-7](#)
 - JSON schema, [5-40](#)
 - motivation, [1-7](#)
 - overview, [1-1](#)

- JSON-relational duality views for car-racing
 - example, [3-1](#)
 - creating with GraphQL, [3-8](#)
 - creating with SQL, [3-4](#)
- JSON-relational mapping (JRM), definition, [1-10](#)
- JSON-To-Duality Migrator, [2-1](#)
- JSON-type column DATA, for duality-view documents, [3-1](#)
- JSON-type payload column DATA, supported/generated by a duality view, [3-1](#), [5-38](#)

L

- lock-free (optimistic) concurrency control, [5-23](#)
 - definition and overview, [4-3](#)
 - duality-view transactions, [5-32](#)

M

- many-to-many entity relationships, [2-6](#)
 - using mapping tables, [2-8](#)
- many-to-one entity relationships, [2-6](#)
- mapping objects/documents to relational, [1-10](#)
- mapping table for tables driver and race, [2-8](#)
- mapping table, definition, [2-8](#)
- merging flex-column fields into documents, [8-1](#), [8-4](#), [8-8](#)
- migration of document sets to duality views, [2-1](#)
- MongoDB API, compatible document-identifier field _id, [6-1](#)
- multitenant database, definition, [1-13](#)

N

- N:N entity relationships, [2-6](#)
 - using mapping tables, [2-8](#)
- naming conflicts, flex column, [8-8](#)
- NEST SQL keyword, [3-4](#)
- NOCHECK annotation (ETAG calculation), [4-3](#)
- NODELETE annotation, [4-2](#)
- NOINSERT annotation, [4-2](#)
- normalization, degree/granularity, [8-6](#)
- normalized data, definition, [1-7](#)
- normalized entity, definition, [2-6](#)
- NOUPDATE annotation, [4-2](#)

O

- object-document mapping (ODM), [1-10](#)
- object-relational mapping (ORM), [1-10](#)
- ODM (object-document mapping), [1-10](#)
- one-to-one entity relationships, [2-6](#)
- operations on duality views, [5-1](#)
- operations on tables underlying duality views, [5-1](#)

optimistic (lock-free) concurrency control, [5-23](#)
 definition and overview, [4-3](#)
 duality-view transactions, [5-32](#)
 optimization of document operations, [5-38](#)
 Oracle Database API for MongoDB, compatible
 document-identifier field `_id`, [6-1](#)
 Oracle REST Data Services (ORDS)
 deleting documents using REST, [5-10](#)
 inserting documents using REST, [5-3](#)
 updating documents using REST, [5-13](#)
 Oracle SQL function `json_transform`, [5-13](#)
 ORM (object-relational mapping), [1-10](#)

P

payload JSON-type column DATA, supported/
 generated by a duality view, [3-1](#), [5-38](#)
 payload of a JSON document, definition, [2-2](#), [3-1](#),
 [5-23](#)
 PL/SQL subprograms
 DBMS_JSON_DUALITY.begin_transaction,
 [5-32](#)
 DBMS_JSON_DUALITY.commit_transaction,
 [5-32](#)
 DBMS_JSON_DUALITY.register, [5-32](#)
 polyglot database, definition, [1-13](#)
 predefined fields for duality views
 See fields
 pretty-printing
 in book examples, [xi](#)
 primary key, definition, [2-8](#)
 privileges needed for operations on duality-view
 data, [4-5](#)

Q

querying a duality view, [5-38](#)

R

race and driver mapping table, [2-8](#)
 race document, [2-2](#)
 race duality view, [3-1](#)
 creating with GraphQL, [3-8](#)
 creating with SQL, [3-4](#)
 JSON schema, [5-40](#)
 race entity, [2-6](#)
 race table, [2-8](#)
 read consistency, ensuring, [5-36](#)
 relational data as starting point, [1-4](#)
 relational mapping from objects/documents, [1-10](#)
 RESID hidden duality-view column for document
 identifier, [3-1](#), [5-32](#)
 REST
 deleting documents using, [5-10](#)
 inserting documents using, [5-3](#)

REST (*continued*)
 updating documents using, [5-13](#)
 reusing existing relational data, for JSON
 documents, [1-4](#)
 row-identifier field, `_id`, [6-1](#)
 rules for updating duality views, [4-6](#)

S

schema evolution, [8-8](#)
 schema flexibility, [8-2](#), [8-4](#), [8-6](#), [8-8](#)
 schema, JSON
 description of duality view, [5-40](#)
 use to validate JSON-column data, [1-7](#), [2-8](#),
 [5-13](#), [8-2](#), [8-6](#), [8-8](#)
 SCN
 See system change number
 secondary key, [2-8](#)
 security, [1-12](#)
 sharing JSON data among documents, [1-7](#), [8-1](#)
 foreign keys, [2-8](#)
 SQL function `json_transform`, [5-13](#)
 SQL/JSON function `json_value`, RETURNING
 clause, used to optimize operations, [5-38](#)
 SQL/JSON generation functions, [3-1](#)
 SQL/JSON item methods, used to optimize
 operations, [5-38](#)
 static dictionary views for duality views, [5-40](#)
 storing JSON data in underlying tables, [8-1](#), [8-2](#),
 [8-6](#)
 support of document collection by duality view,
 definition, [1-1](#)
 support of documents by a duality view, definition,
 [1-1](#)
 SYS_ROW_ETAG function, optimistic
 concurrency control, [5-23](#)
 system change number (SCN) field, asof, [3-1](#)
 ensuring read consistency, [5-36](#)

T

table operations, effect on supported documents,
 [5-1](#)
 table-centric use of duality views, [1-4](#)
 tables
 car-racing example, [2-8](#)
 deleting data, [5-10](#)
 inserting data, [5-3](#)
 updating data, [5-13](#)
 tables underlying duality views, column types, [2-8](#)
 team document, [2-2](#)
 team duality view, [3-1](#)
 creating with GraphQL, [3-8](#)
 creating with SQL, [3-4](#)
 JSON schema, [5-40](#)
 team entity, [2-6](#)

team table, [2-8](#)
transactions for duality views, [5-32](#)
triggers, guidelines, [5-22](#)
type-conversion item methods, used to optimize
 operations, [5-38](#)
types of columns in tables underlying duality
 views, [2-8](#)

U

unique key, definition, [2-8](#)
UNNEST SQL keyword, [3-4](#)
updatability, defining, [4-2](#)
UPDATE annotation, [4-2](#)
updating documents, [5-13](#)
updating duality views
 privileges needed, [4-5](#)
 rules, [4-6](#)
USER_JSON_DUALITY_VIEW_TAB_COLS view,
 [5-40](#)

USER_JSON_DUALITY_VIEW_TABS view, [5-40](#)
USER_JSON_DUALITY_VIEWS view, [5-40](#)

V

value-based ETAG concurrency control, definition,
 [5-23](#)
version-identifier field, etag, [3-1](#)
 controlling concurrency, [5-23](#)
view, duality
 See duality view
views for JSON data, dictionary, [1-7](#)
views, static dictionary, [5-40](#)

W

WHERE clauses, duality-view tables, [3-14](#)