

# Oracle Rdb™

A Comparison of SQL Dialects for Oracle and Oracle Rdb

Release 1.0

April 8, 1997

Part No. A53248-01

---

A Comparison of SQL Dialects for Oracle and Oracle Rdb

Part No. A53248-01

Release 1.0

Copyright © 1997, Oracle Corporation. All rights reserved.

**The programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.**

This Program contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright patent and other intellectual property law. Reverse engineering of the software is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free.

If this Program is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

**Restricted Rights Legend** Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Oracle, SQL\*Net, and SQL\*Plus are registered trademarks of Oracle Corporation, Redwood City, California.

Oracle7, Oracle Call Interface, Oracle Rdb, Oracle RMU, Oracle7 Server, DBAPack, Developer/2000, PL/SQL, Pro\*C, and Rdb7 are trademarks of Oracle Corporation, Redwood City, California.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

---

---

# Contents

<b>Send Us Your Comments .....</b>	<b>vii</b>
<b>Preface.....</b>	<b>ix</b>
<b>1 Data Model</b>	
1.1 Schemas.....	1-1
1.1.1 Tables .....	1-3
1.1.2 Constraints .....	1-3
1.1.3 Domains.....	1-5
1.1.4 Sequences .....	1-5
1.1.5 Synonyms .....	1-6
1.1.6 Triggers.....	1-6
1.1.7 Views.....	1-7
1.2 Database Link .....	1-8
1.3 Nonschema Objects.....	1-9
1.4 Distributed Data .....	1-9
1.5 Namespaces.....	1-10
1.6 Data Types.....	1-11
1.6.1 Character Data Types .....	1-11
1.6.2 Numeric Data Types.....	1-12
1.6.3 Date-Time Data Types.....	1-14
1.6.4 Bit String Data Types.....	1-15
1.6.5 Long Data Types .....	1-15
1.6.6 Other Data Types .....	1-17
1.7 Multinational Character Sets .....	1-17

## 2 Data Operations

2.1	Operators .....	2-1
2.2	Concatenation .....	2-2
2.3	Conversions.....	2-2

## 3 Lexical Issues

3.1	SQL Names.....	3-1
3.2	Qualified Names.....	3-2
3.2.1	Qualified Table Names.....	3-2
3.2.2	Qualified Column Names .....	3-3
3.3	Comment Indicators.....	3-3
3.4	“Not equals” Operators.....	3-4
3.5	Outer Join (+).....	3-4
3.6	Reserved Words and Keywords.....	3-5

## 4 Query Specifications

4.1	Pseudocolumns.....	4-1
4.1.1	Row Addresses (ROWID and DBKEY) .....	4-1
4.1.2	Number of Rows Returned (ROWNUM) .....	4-2
4.1.3	Values of Sequences (NEXTVAL and CURRVAL) .....	4-3
4.1.4	Recursive Unions (LEVEL) .....	4-3
4.2	Functions.....	4-4
4.2.1	DECODE Function .....	4-11
4.2.2	CASE Expression.....	4-12
4.3	Predicates.....	4-12
4.4	Select Expression .....	4-14
4.4.1	SELECT List.....	4-15
4.4.2	Derived Tables .....	4-15
4.4.3	Scope of Table Names.....	4-16
4.4.4	ORDER BY Clause.....	4-17
4.4.5	GROUP BY Clause .....	4-18

## 5 Data Manipulation

5.1	Database Attachments .....	5-1
-----	----------------------------	-----

5.2	Subqueries (UPDATE, DELETE and INSERT) .....	5-2
5.3	UPDATE Statement.....	5-2
5.3.1	Set Single or Multiple Columns (SET Clause).....	5-2
5.3.2	Mix Values and Subqueries (UPDATE and INSERT Statements) .....	5-3
5.3.3	Bulk Updates on a Column with a UNIQUE Constraint .....	5-4
5.3.4	Correlated Subqueries .....	5-5
5.4	DELETE [FROM] Table .....	5-5

## 6 Data Definition Statements

6.1	Data Definition Language and Transactions.....	6-1
6.2	Data Type Usage (CREATE TABLE, CREATE DOMAIN, Declare Variable).....	6-1
6.2.1	Data Type Synonyms.....	6-2
6.2.2	Data Type Defaults .....	6-2
6.3	Table Creation (CREATE TABLE ... AS SELECT) .....	6-3
6.4	Table Alterations (ALTER TABLE).....	6-4
6.5	Constraints .....	6-4
6.5.1	Referential Delete (ON DELETE CASCADE) .....	6-4
6.5.2	Unique Indexes (UNIQUE or PRIMARY KEY) .....	6-5
6.5.3	Constraint Exceptions (EXCEPTIONS INTO).....	6-5
6.5.4	Disabled and Enabled Constraints (DISABLE, ENABLE) .....	6-5
6.5.5	Deferred Constraints .....	6-6
6.6	CREATE OR REPLACE Clause for Compiled Objects.....	6-6
6.7	CREATE VIEW Statement.....	6-7
6.8	Add Comments to Describe Metadata or Data Definitions (COMMENT ON) .....	6-7
6.9	Delete Definitions (DROP ... CASCADE) .....	6-8
6.10	Object Recompile (ALTER FUNCTION, ALTER TRIGGER, ALTER VIEW, ALTER PROCEDURE) 6-8	
6.11	Rename Database Objects (RENAME).....	6-9

## A Reserved Words

### Index

## Tables

1-1	Oracle7 and Rdb7 SQL Character Data Types .....	1-11
1-2	Oracle7 and Rdb7 Numeric Data Types .....	1-12
1-3	Oracle7 and Rdb7 SQL Date-Time Data Types .....	1-14
1-4	Oracle7 and Rdb7 SQL Bit String Data Types .....	1-15
1-5	Oracle7 and Rdb7 SQL Long Data Types.....	1-15
1-6	Other Oracle7 and Rdb7 Data Types .....	1-17
4-1	Functions Supported by Oracle7 and Rdb7 .....	4-4
6-1	Data Type Defaults for Oracle7 and Rdb7 SQL .....	6-2
A-1	Comparison of Reserved Words and Semantics .....	A-1

---

---

# Send Us Your Comments

## A Comparison of SQL Dialects for Oracle and Oracle Rdb, Release 1.0

Part No. A53248-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- electronic mail - [nedc\\_doc@us.oracle.com](mailto:nedc_doc@us.oracle.com)
- FAX - 603-897-3334. Attn: Oracle Rdb Documentation
- postal service  
Oracle Corporation  
Oracle Rdb Documentation  
One Oracle Drive  
Nashua, NH 03062  
USA.

If you would like a reply, please give your name, address, and telephone number below.

---

---

---





---

---

# Preface

This manual compares the SQL dialects of Oracle7 and Rdb7, and describes the known differences between them.

## Purpose of This Manual

The principal purpose of this manual is to help users of the SQL\*Net for Rdb software understand differences in the Oracle7 and Rdb7 SQL dialects. In general, SQL\*Net for Rdb recognizes most SQL statements and translates them transparently. However, in some cases, SQL\*Net for Rdb cannot exactly interpret some SQL statements.

This manual identifies where differences in the SQL dialects might occur, and the discussions in this manual provide additional information to help you achieve the desired functions.

The topics in this manual are restricted to the logical schema and operations performed on the physical data.

This manual does not describe:

- Physical storage, physical representation, or optimization issues
- Compiler, precompiler, and interactive SQL features
- PL/SQL
- Data dictionary features
- Features common to Oracle7 and Rdb7

## Intended Audience

This manual is intended primarily for experienced SQL programmers who need to access data in an Rdb7 database using SQL\*Net for Rdb. The topics in this manual can also be helpful to the following:

- Oracle7 programmers writing applications to access data on Rdb7 databases
- Rdb7 programmers using Oracle7 client tools (such as Developer/2000 or PL/SQL) to access data on Rdb7 databases
- Programmers developing applications to run in both environments

To use this manual most effectively, you need to be familiar with Rdb7 database management concepts and terminology, Oracle7 SQL statements, or Rdb7 SQL statements.

The Oracle7 and Rdb7 SQL dialects both comply with the ANSI/ISO SQL Entry-Level Standard, commonly referred to as SQL92. Oracle7 SQL has one fixed dialect. The dialect for Rdb7 SQL is Oracle Level1.

## Structure

This manual contains six chapters, an appendix, and an index:

---

Chapter 1	Compares data models for Oracle7 and Rdb7.
Chapter 2	Compares data operations for Oracle7 and Rdb7.
Chapter 3	Compares lexical operations for Oracle7 and Rdb7.
Chapter 4	Compares query specifications for Oracle7 and Rdb7.
Chapter 5	Compares data manipulation for Oracle7 and Rdb7.
Chapter 6	Compares data definition statements for Oracle7 and Rdb7.
Appendix A	Provides a comparison of reserved words and semantics for the Oracle7 and Rdb7 SQL dialects.

---

## Organization

Each chapter and the appendix identify common elements and differences in SQL semantics between the Oracle7 and Rdb7 environments. To make it easier for you to pick out information relevant to you, each section is divided into the following discussions:

- **Common**—Describes characteristics and functions that are common to both the Oracle7 and Rdb7 SQL dialects.

In general, because both Oracle7 and Rdb7 support the SQL92 standard, these basic features will not be described except to the extent of differences in interpretation. These features are described in both Oracle7 and Rdb7 documentation, as well as in the ANSI/ISO SQL standard.

- **Oracle7**—Describes characteristics specific to Oracle7 SQL.
- **Rdb7**—Describes characteristics specific to Rdb7 SQL.

Wherever possible, when a topic is specific to one SQL dialect, the discussion will provide additional information to help you achieve the same or similar function from the other SQL dialect.

## Related Documents

The following list provides the names of documents to which you can refer for supplemental information:

- For information about the Rdb7 product and using the Rdb7 SQL statements, refer to the following documentation:
  - *Oracle Rdb Release Notes*
  - *Oracle Rdb SQL Reference Manual*
- For information about the Oracle7 SQL commands and functions:
  - *Oracle7 Server SQL Language Reference Manual*
  - *PL/SQL User's Guide and Reference*
  - *SQL\*Plus User's Guide and Reference*
- For information about using the Oracle Call Interface (OCI) to build Oracle Developer/2000 applications that access data in Rdb7 databases, refer to the *Programmer's Guide to the Oracle Call Interface*.
- For more information about using the SQL\*Net for Rdb software to develop transparent connections between SQL\*Net clients and Oracle Rdb databases, refer to the *Oracle Rdb Guide to SQL\*Net for Rdb*.
- For information about building Oracle Developer/2000 applications that access data in Rdb7 databases, refer to:
  - The supplemental documentation addendum of the *Oracle Developer/2000 Forms Advanced Techniques Manual*

- The *Oracle Developer/2000 Installation and User's Guide* (Release 1.3)
- For more information regarding the SQL standard, refer to ANSI X3.135-1992, ISO 9075:1992.

## Conventions

- In this manual, OpenVMS refers to both the OpenVMS Alpha and the OpenVMS VAX operating systems.
- Rdb7 refers to Oracle Rdb Release 7.0 for OpenVMS
- Oracle7 refers to Oracle Release 7.0 (and higher) software.
- The Oracle Rdb SQL server is referred to as Rdb7 SQL.
- The Oracle7 SQL server is referred to as Oracle7 SQL.
- Oracle Developer/2000 is referred to as Developer/2000.
- Oracle PL/SQL is referred to as PL/SQL.
- The SQL standard, ANSI X3.135-1992, ISO 9075:1992, is referred to as either the ANSI/ISO SQL standard or SQL92.
- The SQL standard, ANSI X3.135-1989, ISO 9074:1989, is referred to as SQL89.
- The next proposed SQL standard, which is still under development, is referred to as SQL3.
- In examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the Return key at the end of a line of input.

The following conventions are also used in this manual:

Convention	Meaning
. . . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
...	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted.
<b>boldface text</b>	Boldface type in text indicates a term defined in the text.

---

<i>italic text</i>	Italic text emphasizes important information, information supplied by the user, titles of manuals, and variables. Variables include information that varies in system messages (Internal error <i>number</i> ), in command lines ( <i>/PRODUCER=name</i> ), and in command parameters in text (where <i>device-name</i> is the current disk name).
[ ]	Brackets enclose optional clauses from which you can choose one or none.
\$	The dollar sign represents the DIGITAL Command Language prompt for OpenVMS systems and the Bourne shell prompt for Digital UNIX systems.
[ <i>nn</i> ]	<i>nn</i> , enclosed in brackets, indicates an optional product version number for a file name in a multiversion environment. If you omit the version number, you are using the standard version environment.

---



This chapter identifies common elements and differences for SQL data models.

## 1.1 Schemas

### **Common**

The following objects serve similar functions in both Oracle7 and Rdb7 SQL schemas:

- Tables, including:
  - Columns
  - Constraints
  - Indexes
  - Pseudocolumns
- Views
- Stored modules, containing stored routines
- External routines

In addition, the following objects in Oracle Procedural Option or Distributed Option have counterparts in Rdb7:

- Database triggers
- Standalone procedures and stored functions

---

---

**Note:** Even though Oracle7 procedures, triggers, and functions have counterparts in Rdb7, Oracle7 uses these objects in a PL/SQL procedure, which is considerably different from Rdb7 SQL. The contents of procedural objects must be written separately for Oracle7 and Rdb7.

---

---

SQL\*Net for Rdb does not support multischema databases.

### **Oracle7**

All Oracle7 databases are multischema databases in which each user has ownership of a schema that is named with the user's user name. Therefore, with Oracle7 SQL, you can create a schema with only the user name or the user ID. However, even though each schema is individually owned and named for a user, Oracle7 SQL allows users to access objects in other users' schemas by means of synonyms or explicit reference.

The following are objects in Oracle7 SQL schemas:

- Sequences
- Synonyms
- Database links

The following objects are available with the Oracle7 Procedural Option or Distributed Option:

- Packages
- Snapshots
- Snapshot logs

### **Rdb7**

In contrast to Oracle7 databases, you can create a multischema database with Rdb7 SQL, which allows many schemas per user whose names are not restricted. However, Rdb7 SQL is usually used in single-schema mode. The following objects are specific to Rdb7 SQL schemas:

- Domains
- Collating sequences
- Tables, including computed columns
- Temporary tables



- Outlines

The following objects are specific to Distributed Option for Rdb:

- Transfers
- Schedules

## 1.1.1 Tables

### Common

In general, tables are similar in both the Oracle7 and Rdb7 SQL dialects.

### Oracle7

The use of the CREATE AS SELECT statement is specific to Oracle7. When using the CREATE AS SELECT statement, you can populate an Oracle7 table at the time it is created. This is similar to the combination of a CREATE TABLE *table-name* statement and an INSERT INTO *table-name* SELECT... FROM statement.

The CREATE AS SELECT statement is executed as an atomic unit. Therefore, if the insertion of data fails, the table is not created.

### Rdb7

The following objects are specific to Rdb7 SQL:

- Computed columns

Rdb7 SQL supports read-only columns that contain derived data, specified by the COMPUTED BY clause. The COMPUTED BY clause can contain subqueries referring to other tables in the same database.

- Temporary tables

Rdb7 SQL supports both local and global temporary tables, as defined in the ANSI/ISO SQL standard. In particular, the data held in these tables is specific to the session and the table may have been created in such a way that committing a transaction deletes its data.

## 1.1.2 Constraints

### Common

The following constraints are comparable in both the Oracle7 and Rdb7 SQL dialects:

- CHECK
- FOREIGN KEY
- NOT NULL
- PRIMARY KEY
- UNIQUE
- NULL (default behavior)

### Oracle7

The following are Oracle7 SQL constraints:

- FOREIGN KEY... CASCADE

Oracle7 SQL supports deleting the referencing rows when the referenced row is deleted, as in FOREIGN KEY *foreign-key* REFERENCES *other-table (pk)* ON DELETE CASCADE.

- NULL

Oracle7 SQL allows you to explicitly specify the NULL keyword when creating a table. For example, CREATE TABLE ... *column-name data-type* NULL. This is not a constraint, but merely denies the NULL attribute, so that the column can contain null values. This is the default behavior for both Oracle7 and Rdb7 databases. If your goal is to have common code, avoid using the NULL keyword.

- Oracle7 SQL does not support creating constraints with the DEFERRABLE or DEFERRED attributes.
- In Oracle7 SQL, constraints cannot contain subqueries.
- All Oracle7 constraints are IMMEDIATELY DEFERRED. However, Oracle7 SQL does allow altering a table to disable its constraints.

### Rdb7

The following are Rdb7 SQL constraints:

- CHECK (referring outside the table)

A CHECK constraint can contain a subquery referring to other tables (or the same table).

- CHECK (nondeterministic)

Rdb7 SQL allows constraints to reference system functions that return different results, such as `USER` and `CURRENT_TIME`. Constraints are verified only for rows that have been altered. Otherwise, such tables might not succeed in storing any data. This behavior is different from the ANSI/ISO SQL standard.

- **DEFERRABLE, DEFERRED**

Rdb7 SQL supports creating constraints with `DEFERRABLE`, `NOT DEFERRABLE`, or `DEFERRED` attributes.

- **UNIQUE**

A `UNIQUE` table constraint created prior to Oracle Rdb Release 7.0 allows a maximum of one instance of a `NULL` value. This behavior is different from the ANSI/ISO SQL standard.

The behavior of constraints created in Oracle Rdb Release 7.0 and higher, has been changed to comply with the ANSI/ISO SQL standard.

- **NULL**

With Rdb7 SQL, you cannot explicitly specify the `NULL` keyword as you can with Oracle7 SQL. However, the default behavior is the same as with Oracle7 SQL. If your goal is to have common code, avoid using the `NULL` keyword when using Oracle7 SQL.

### 1.1.3 Domains

**Common**

None.

**Oracle7**

Oracle7 SQL does not support domains.

**Rdb7**

Rdb7 SQL supports domain definitions according to the specifications in the ANSI/ISO SQL standard.

### 1.1.4 Sequences

**Common**

None.

### **Oracle7**

The CREATE SEQUENCE statement is specific to Oracle7 databases. A sequence is a database object from which you can generate integers, typically for primary key values.

### **Rdb7**

Rdb7 SQL does not support sequences.

## **1.1.5 Synonyms**

### **Common**

None.

### **Oracle7**

The Synonyms object and usage is specific to Oracle7 databases.

Oracle7 SQL allows public and private synonyms to be defined for schema objects that are either local to a schema or shared by all users. Such an object can be a:

- Table or view
- Sequence
- Stored routine
- Snapshot
- Synonym

Synonyms provide a convenient way to use an object in a different schema.

The object does not have to exist until the synonym is used.

### **Rdb7**

Rdb7 SQL does not support synonyms, however, creating a view usually meets the same needs.

## **1.1.6 Triggers**

### **Common**

The triggers supported by Oracle7 SQL and Rdb7 SQL are similar. Only the triggered action itself differs.

**Oracle7**

Oracle7 SQL allows a more general PL/SQL block. Oracle7 SQL requires the procedural option in order to define triggers. In addition, you can enable and disable Oracle7 SQL triggers.

**Rdb7**

In Rdb7 SQL, the allowed triggered actions include INSERT, UPDATE, DELETE, and ERROR.

## 1.1.7 Views

**Common**

Views are similar in both the Oracle7 and Rdb7 SQL dialects.

**Oracle7**

The following items are specific to Oracle7 SQL views:

- Oracle7 SQL allows you to create a view with a FORCE option (CREATE FORCE statement). That is, even if the underlying tables do not exist, Oracle7 creates the view and marks it as invalid. The view definition is checked and compiled the next time it is referenced.
- A view in Oracle7 SQL that selects the ROWID of one of its underlying tables must rename the column either by naming all its columns or by assigning a correlation name to the column in the select list.
- A view in Oracle7 SQL that selects the ROWNUM of one of its underlying tables must rename the column either by naming all its columns or by assigning a correlation name to the column in the select list.
- Oracle7 SQL may allow some joined expressions to be updated. For example, expressions that can be updated might include those that have a unique index on a join column. (See the Oracle7 SQL documentation for more information.)

**Rdb7**

The following items are specific to Rdb7 SQL views:

- A view in Rdb7 SQL can use either the ROWID or DBKEY of its underlying tables, with no restrictions.
- A view in Rdb7 SQL does not permit selecting or using ROWNUM in a view definition. For example:

```
CREATE VIEW vr AS SELECT * FROM t1 WHERE ROWNUM < 2
SQL-F-NOTROWNUM ROWNUM is not available in this context.
```

The LIMIT TO clause is similar in function to ROWNUM and can appear in a view definition.

- Rdb7 SQL does not allow updating of a joined expression.
- Even with the dialect set to SQL92, Rdb7 SQL does not check ANSI update rules. For instance, you can update a view that:
  - Uses a SELECT DISTINCT clause

```
CREATE VIEW vd AS SELECT DISTINCT x,y FROM tab;
```

An UPDATE *vd* statement updates an arbitrarily selected set of rows of *tab* chosen to represent the values of the pair (*x*, *y*).

- Selects the same column more than once:

```
CREATE VIEW vm (a, b, c) AS SELECT x, y, x FROM tab;  
UPDATE vm SET a = 1, b = 2, c = 3)
```

The UPDATE *vm* statement performs the first update, setting column *x* to 1, and later writes over that setting *x* to 3.

This behavior is consistent with that of the Oracle7 SQL dialect.

## 1.2 Database Link

### Common

None.

### Oracle7

With Oracle7 SQL, you can use a database link as an object in your local database to access data in other Oracle or non-Oracle databases. For example:

```
CREATE PUBLIC DATABASE LINK personnel_west  
CONNECT TO smith IDENTIFIED BY xyyzz USING ... ;  
SELECT * FROM employees UNION SELECT * FROM employees@personnel_west
```

The CREATE DATABASE LINK statement provides read-only access to foreign data. To gain read/write access, use Oracle Distributed Option.

**Rdb7**

While Rdb7 SQL does not have a database link object, you can create a database alias.

Rdb7 SQL statements cannot access data from more than one database within the same statement. However, Distributed Option for Rdb provides Rdb7 SQL with capabilities that are similar to the Oracle7 SQL database link. The Distributed Option for Rdb product imports all metadata into one virtual database using the CREATE LINK statement. However, SQL\*Net for Rdb does not support Distributed Option for Rdb.

## 1.3 Nonschema Objects

**Common**

None.

**Oracle7**

The following list shows Oracle7 SQL objects in the database that are not contained in schemas:

- Profiles

Oracle7 SQL supports the definition of profile objects that limit database resources.

- Roles

The roles object defines a set of privileges that can be granted, as a whole, to users or to other roles. This is an SQL3 feature supported by Oracle7 SQL.

- Users

A created user may be assigned a profile.

**Rdb7**

Rdb7 SQL does not support nonschema objects. However, using interactive SQL, you can limit the resources used by queries with the SET QUERY statement.

## 1.4 Distributed Data

**Common**

None.

The capabilities (but not the implementation) of the Oracle7 Snapshot object and the Snapshot Log database object correspond to that of the Rdb7 Transfers object and the Schedules object that are available with the Replication Option for Rdb software.

Distributed data cannot be managed by SQL\*Net for Rdb.

### **Oracle7**

Oracle7 SQL (with Oracle Distributed Option):

- Keeps a snapshot copy of the data in a table in an Oracle7 SQL Snapshot database object and logs changes to the original table data in a Snapshot Log database object.
- Both the Snapshot and Snapshot Log database objects are specific to Oracle7 databases.

The definition of the Snapshot includes the procedures needed to refresh the data.

### **Rdb7**

Rdb7 SQL (with Replication Option for Rdb):

- Transfer definitions specify which tables and views are to be replicated into which database and details on the method of transfer.
- Schedule definitions determine how often to replicate the data as specified in a transfer definition.

See the Replication Option for Rdb documentation for more information.

## **1.5 Namespaces**

### **Common**

None.

### **Oracle7**

The namespace of tables and views is shared with sequences, private synonyms, standalone procedures and functions, packages, and snapshots.



**Rdb7**

The namespace of tables and views is not shared with any other objects, although stored procedures and functions share a namespace with external procedures and functions.

## 1.6 Data Types

The following sections compare character, numeric, date-time, bit string, and long data types for the Oracle7 and Rdb7 SQL dialects.

See Section 6.2 for more information on data type usage.

### 1.6.1 Character Data Types

Table 1–1 shows the comparison between Oracle7 and Rdb7 character data types.

**Table 1–1 Oracle7 and Rdb7 SQL Character Data Types**

Oracle7	Description/Key	Rdb7
CHAR (s) 0 < s ≤ 2000 See Note 1 that follows.	Fixed-length character string n = number characters c = number bytes per character s = total number byte = n*c	CHAR (n) 0 < s < 65535 See Notes 1 and 2 that follow.
VARCHAR2 (s) 0 < s ≤ 2000 See Note 3 that follows.	Varying-length character string.	VARCHAR (n) 0 < s ≤ 65535 See Notes 1 and 2 that follow.
VARCHAR (s) 0 < s ≤ 2000 See Note 4 that follows.		

**Notes:**

1. Oracle7 and Rdb7 SQL support several different character sets, including some multi-octet character sets.

In Oracle7 SQL, the length given in the data type is the number of bytes needed to store the data. In Rdb7 SQL, the length given in the data type is the number of characters. The length of a character string, in number of characters, is returned by the LENGTH function for Oracle7 SQL or the CHARACTER\_LENGTH function for Rdb7 SQL. The number of bytes is returned by the LENGTHB function for Oracle7 SQL or the OCTET\_LENGTH function for Rdb7 SQL.

2. These maximum lengths can be used for character parameters. But they cannot be realized for columns in a table because the maximum size of a stored row is 65,272 bytes, which includes the size of all included columns as well as some overhead.
3. When an element of type VARCHAR2 is compared with a character string, no extra blanks are added, but any original trailing blanks are preserved. Rdb7 SQL, following the ANSI/ISO SQL standard, ignores trailing blanks. Therefore, 'string' and 'string ' are the same to Rdb7 SQL but different to Oracle7 SQL. This affects not only the direct comparisons (=, <, and >), but also the results of MAX, MIN, GROUP BY, ORDER BY, SELECT DISTINCT, UNION (DISTINCT), UNIQUE constraints, and equijoins.
4. The Oracle7 SQL data type, VARCHAR (s), is currently synonymous with the Rdb7 SQL data type VARCHAR (n).
5. Oracle7 SQL treats a zero-length string as the null value, behavior that is different from the ANSI/ISO SQL standard. Rdb7 SQL emulates this behavior in its Oracle Level1 dialect.

## 1.6.2 Numeric Data Types

Table 1-2 shows the comparison between Oracle7 and Rdb7 SQL numeric data types.

**Table 1-2 Oracle7 and Rdb7 Numeric Data Types**

Oracle7	Description/Key	Rdb7
NUMBER (p,s) 1 <= p <= 38 decimal precision -84 <= s <= 127	Fixed-point, signed numbers: <ul style="list-style-type: none"> <li>▪ Decimal precision <i>p</i> = number of significant digits</li> <li>▪ Scale <i>s</i> = number of digits to the right of decimal point</li> <li>▪ Note: integers are a special case where <i>s</i> = 0.</li> </ul>	NUMERIC (p,s) DECIMAL (p,s) 1 <= p <= 32767 0 <= s <= 127 See Note 2 that follows.

Table 1–2 Oracle7 and Rdb7 Numeric Data Types (Cont.)

Oracle7	Description/Key	Rdb7
	Fixed-point, signed numbers:	$0 \leq s \leq 127$
	<ul style="list-style-type: none"> <li>▪ Binary precision <math>p</math>, scale <math>s</math></li> </ul>	TINYINT ( $s$ ) *binary* precision 8 $-128 \leq val \leq 127$  SMALLINT ( $s$ ) *binary* precision 16 $-32768 \leq val \leq 32767$  INTEGER ( $s$ ) *binary* precision 32 $2^{**31} \leq val \leq 2^{**31} - 1$  BIGINT ( $s$ ) *binary* precision 64 $2^{**63} \leq val \leq 2^{**63} - 1$
FLOAT ( $p$ ) $0 \leq p \leq 126$	Floating point, precision $p$	FLOAT ( $p$ ) $1 \leq p \leq 53$
NUMBER decimal precision 38 See Note 3 that follows.		REAL 32-bit float, binary precision 24  DOUBLE PRECISION 64-bit float, binary precision 53

**Notes:**

1. Oracle7 SQL supports a negative scale. Rdb7 SQL does not support negative scale.
2. Rdb7 SQL converts DECIMAL and NUMERIC data types to SMALLINT, INTEGER, BIGINT, or FLOAT data types according to whether  $p \leq 4$ ,  $4 < p \leq 9$ ,  $0 < p \leq 18$ , or  $18 < p$ . However, the data is constrained to have a value with decimal precision  $\leq p$ . An informational message may be displayed about the change in format. If the number is converted to FLOAT, some precision may be lost and no informational message will be displayed.  
  
DECIMAL represents packed decimal; NUMERIC represents signed numeric.
3. Oracle7 SQL maps the ANSI/ISO SQL standard data type INTEGER to NUMBER, which has decimal precision 38. However, if more significant digits are

specified, no error is reported and accuracy may be lost. The actual precision retained is platform dependent. For example:

```
Oracle_SQL> CREATE TABLE ti (x INTEGER);
Oracle_SQL> INSERT INTO ti (x) VALUES (1234567890123456789012345678901234567896123456);
/* 46 digits */
Oracle_SQL> SELECT * FROM ti;
      X
-----
1.2346E+45
Oracle_SQL> SELECT * FROM ti WHERE x > 1234567890123456789012345678901234567896123451;
/* lower the last digit, expect 1 row back */
no rows selected
Oracle_SQL> SELECT * FROM ti WHERE x = 1234567890123456789012345678901234567896000000;
/* precision has been truncated to 40 */
      X
-----
1.2346E+45
```

- The format in which a fixed-point number is presented, by Oracle7 SQL, to the user may appear to have lost precision. To fit the number into the assigned column width, if it has too many digits, Oracle7 SQL will represent it in floating-point notation, rounded to however many digits will fit within the assigned column width. The original precision is preserved.

### 1.6.3 Date-Time Data Types

Table 1-3 shows the comparison between Oracle7 and Rdb7 SQL date-time data types.

**Table 1-3 Oracle7 and Rdb7 SQL Date-Time Data Types**

Oracle7	Description/Key	Rdb7
DATE format is controlled by date-format	Full date and time (Rdb7 supports from year to fraction of second.)  Date (year to day)	DATE (See Note 1 that follows.) DATE VMS  DATE ANSI
	Time, hour to fraction of second $f$ = precision of fractional part	TIMESTAMP ( $f$ ) $0 \leq f \leq 2$  TIME ( $f$ ) $0 \leq f \leq 2$

**Table 1–3 Oracle7 and Rdb7 SQL Date-Time Data Types (Cont.)**

Oracle7	Description/Key	Rdb7
	Interval, year-month or day-time <i>f</i> = precision of fractional part	INTERVAL ( <i>f</i> ) <i>f</i> is valid only if seconds are included

**Notes:**

1. DATE, in Rdb7 SQL, will be ANSI or VMS according to the DEFAULT DATE FORMAT. The format used for data stored in a table is set by the DEFAULT DATE FORMAT value when the table was created. For the Oracle Level1 dialect, the default date format is DATE VMS. For the ANSI/ISO SQL standard, the default date format is DATE ANSI. The CAST operator can convert dates from one format to the other.
2. Valid dates in Oracle7 SQL range from 1 Jan 4712 BC to 31 Dec 4712 AD. Valid dates in Rdb7 SQL range from 1 Jan 1859 to 31 Dec 9999.

## 1.6.4 Bit String Data Types

Table 1–4 shows the comparison between Oracle7 and Rdb7 SQL bit string data types.

**Table 1–4 Oracle7 and Rdb7 SQL Bit String Data Types**

Oracle7	Description/Key	Rdb7
RAW ( <i>n</i> ) 1 <= <i>n</i> <= 255	Variable-length bit string	No equivalent. Use VARCHAR ( <i>n</i> ).

## 1.6.5 Long Data Types

Table 1–5 shows the comparison between Oracle7 and Rdb7 SQL long data types.

**Table 1–5 Oracle7 and Rdb7 SQL Long Data Types**

Oracle7	Description/Key	Rdb7
LONG Maximum length of 2 gigabytes	Variable-length character string	LIST OF BYTE VARYING ( <i>n</i> ) <i>n</i> = segment size 1 <= <i>n</i> <= 65535 No maximum size for data

**Table 1–5 Oracle7 and Rdb7 SQL Long Data Types (Cont.)**

Oracle7	Description/Key	Rdb7
LONG RAW Maximum length of 2 gigabytes	Variable-length bit string	

**Notes:**

1. The LIST OF BYTE VARYING data type (also known as segmented strings) stores the data in a list of segments. Such data cannot be manipulated by the usual INSERT, SELECT, and UPDATE ... SET statements, but is maintained by using list cursors. SQL\*Net for Rdb allows inserts.
2. Oracle7 SQL permits, at most, one LONG or one LONG RAW column in an underlying table.
3. The value of a LONG column can be manipulated with the usual SELECT, INSERT, and UPDATE statements.
4. Both Oracle7 and Rdb7 SQL have many restrictions on LONG and segmented string data:
  - The only valid predicate is IS [NOT] NULL.
  - They cannot be indexed.
  - The only constraint on them is NOT NULL.
  - They cannot be passed to SQL functions.
  - Stored functions cannot return them as a value.
  - They cannot be compared, either explicitly or implicitly, using the ORDER BY, GROUP BY, or SELECT DISTINCT clauses.
  - They cannot occur in expressions, except in a CASE expression for Rdb7 SQL.

In addition, Rdb7 SQL has the following restrictions:<sup>1</sup>

- Literals cannot be inserted into a column defined with the LIST OF BYTE VARYING data type.

---

<sup>1</sup>SQL\*Net for Rdb allows selecting columns defined with the LIST OF BYTE VARYING data type and allows inserting data using a parameter marker into columns defined with the LIST OF BYTE VARYING data type (features that are typically only available with list cursor syntax).

- A column defined with the LIST OF BYTE VARYING data type cannot be updated except with the NULL value, for example, UPDATE TABLE SET *column-name* = NULL.
- Segmented string data cannot be passed as parameters to stored or external routines.

Oracle7 SQL has the following additional restriction:

- If an SQL statement refers to a LONG column, then these columns must be in the same database as all updated and locked tables.

## 1.6.6 Other Data Types

Table 1-6 shows the comparison between other miscellaneous Oracle7 and Rdb7 SQL data types.

**Table 1-6 Other Oracle7 and Rdb7 Data Types**

Oracle7	Description/Key	Rdb7
MLSLABEL	Binary format of a security label (used by Trusted Oracle)	
ROWID	Key to a row in a database	No equivalent external data type. Can use CHAR ( <i>n</i> ).

**Note:**

1. Rdb7 SQL supports the DBKEY keyword for data selected in views and cursors. The value of DBKEY can be represented by CHAR ( $8*n$ ), where *n* is the number of underlying tables from which the data is selected.

## 1.7 Multinational Character Sets

**Common**

None.

**Oracle7**

When you create a database, you set (either implicitly or explicitly) the character set for all character data to be held in the database. This value cannot be changed.

### **Rdb7**

You can independently set the character set for all identifiers, column values, and for literals. The data types NCHAR and NCHAR VARYING refer to character data in the character set appropriate for the object (name, column, or literal). You can also specify a different character set by using the CHARACTER SET IS *character-set-name* clause



---

---

# Data Operations

## 2.1 Operators

The following sections describe Oracle7 and Rdb7 SQL arithmetic and set operators.

### Common

Oracle7 and Rdb7 SQL support the UNION and UNION ALL operators.

The following list shows the order of precedence for arithmetic operations:

1. Unary +, -
2. \*, /
3. Binary +, -, || (concatenation)
4. <, <=, =, >=, >, <>, !=, ^=, !=
5. NOT
6. AND
7. OR

The order of operators is the same for both the Oracle7 and Rdb7 SQL dialects. When operators are at the same level, the precedence is from left to right.

### Oracle7

The Oracle7 SQL operators include INTERSECT and MINUS. These operators are all the same precedence and are evaluated from left to right.

Note the following information about Oracle7 set operators:

- The MINUS operator is equivalent to the EXCEPT operator in the ANSI/ISO SQL standard.

- Oracle7 SQL support for outer joins is different from the language in the ANSI/ISO SQL standard.

### **Rdb7**

Note the following information about Rdb7 set operators:

- The Rdb7 SQL set operators include JOIN (LEFT, RIGHT, FULL, INNER).
- Rdb7 SQL does not support the EXCEPT (MINUS) or INTERSECT set operators.

## **2.2 Concatenation**

### **Common**

Null values are treated as zero-length strings by the Oracle7 SQL dialect and by the Rdb7 SQL Oracle Level1 dialect. This behavior is different from the ANSI/ISO SQL standard.

### **Oracle7**

Oracle7 SQL allows you to concatenate numbers with numbers or with text, implicitly converting them to character data type first.

### **Rdb7**

Rdb7 SQL requires that you perform these functions explicitly using the CAST operator.

## **2.3 Conversions**

### **Common**

Both Oracle7 and Rdb7 SQL convert data automatically between certain data types when storing and retrieving data from the database.

### **Oracle7**

Oracle7 SQL also performs automatic conversions when executing arithmetic expressions.

**Rdb7**

Rdb7 SQL does not automatically convert arithmetic expressions. You must use the CAST operator. However, the Oracle Level1 dialect automatically converts numbers to INTERVAL DAY when performing arithmetic operations with dates.



---

---

## Lexical Issues

This chapter compares Oracle7 and Rdb7 SQL statements with regard to lexical issues.

### 3.1 SQL Names

The following sections provide general information about naming.

#### **Common**

None.

#### **Oracle7**

Oracle7 SQL names follow these rules:

- Object names can include up to 30 characters.
- Database names are limited to 8 characters.
- Database links can have names with up to 128 characters.
- If the name is not enclosed in double quotation marks ("):
  - The name is case insensitive.
  - The at sign (@) character splits the name into an object name and a link.
  - The name must start with an alphabetic character.
  - You can use the following non-alphanumeric characters only within a name:
    - underscore ( \_ )
    - dollar sign ( \$ )
    - pound sign ( # )

### **Rdb7**

Rdb7 SQL names follow these rules:

- Object names can include up to 31 characters.
- Database names must be valid file names as defined by the operating system.
- An alias name (corresponding to a link) is case sensitive on the Digital UNIX platform.
- If the name is not enclosed in double quotation marks ("):
  - The name is case insensitive.
  - The at sign (@) character splits the name into an object name and an alias name if there is a matching alias.
  - The name must start with an alphabetic character.
  - You can use the following non-alphanumeric characters only within a name along with symbols found in the Multinational Character Set:
    - underscore ( \_ )
    - dollar sign ( \$ )
    - percent sign ( % )
    - backslash ( \ )

## **3.2 Qualified Names**

The following sections describe how table names and column names are qualified by the Oracle7 and Rdb7 SQL dialects.

### **3.2.1 Qualified Table Names**

#### **Common**

Both Oracle7 and Rdb7 SQL support the *table-name @ handle* construct to refer to a table in a database with an alias or a database handle.

#### **Oracle7**

Spaces are allowed around the at sign (@) character.

#### **Rdb7**

Spaces are not allowed around the at sign (@) character, so that the combination is treated as one identifier.

For Rdb7 SQL, the form *table-name @handle* sets up *@handle* as a correlation name for the table.

### 3.2.2 Qualified Column Names

#### Common

None.

#### Oracle7

If a column name uses the *table-name @handle* qualification, Oracle7 SQL reports an error.

Oracle7 databases are multischema. However, qualifying a column name with the schema, if its table (in the FROM clause) has not been qualified, produces the following error:

```
SELECT table.t1.x FROM t1;
SELECT table.t1.x FROM t1
                *
ERROR at line 1:
ORA-00904: invalid column name
```

#### Rdb7

If a column name uses the *column-name @handle* qualification, Rdb7 SQL interprets it as a correlation name and an indicator parameter.

If there are no spaces around the at sign (@) character, then Rdb7 SQL treats the whole *column-name@handle* as one name.

## 3.3 Comment Indicators

#### Common

The following list shows the forms of comment demarcations supported by the Oracle7 and Rdb7 SQL dialects:

```
--
/* ... */
```

#### Oracle7

Comment indicators allowed are listed in the Common subheading of this section.

### **Rdb7**

Rdb7 supports the exclamation point (!) as a comment indicator for the SQL interactive interface as well as those listed in the Common subheading of this section.

## **3.4 "Not equals" Operators**

### **Common**

The following list shows the "not equals" operators supported by the Oracle7 and Rdb7 SQL dialects:

```
<>  
!=  
^=
```

### **Oracle7**

Oracle7 SQL also supports the following "not equals" operator:

```
--
```

### **Rdb7**

The "not equals" operators are listed under the Common subheading.

## **3.5 Outer Join (+)**

### **Common**

Both Oracle7 and Rdb7 SQL support implicit joins. An example of achieving this is by listing more than one table in the FROM clause.

### **Oracle7**

The following items pertain to the outer join (+) option for Oracle7:

- An outer join returns all rows that satisfy the join condition and those rows from one table for which no rows from the other satisfy the join condition. A table cannot have an outer join to more than one table in the same SELECT statement. For example:

```
SELECT * FROM employees e, departments d  
WHERE e.dept_id = d.dept_id (+);  
/* left outer join */
```



```
SELECT * FROM employees e, departments d
WHERE e.dept_id (+) = d.dept_id;
      /* right outer join */
```

- Oracle7 SQL uses the syntax *column-name*(+) = *expression* to achieve an outer join, which differs from the ANSI/ISO SQL standard. In the expression, the plus sign indicates that if no match is found, a row of null values is used as a match.

### Rdb7

The following items pertain to the outer join option for Rdb7:

- Rdb7 SQL supports full outer join, left outer join, and right outer join in compliance with the ANSI/ISO SQL standard outer join syntax.
- Rdb7 SQL also supports cross joins and natural joins in compliance with the ANSI/ISO SQL standard join syntax.
- In Rdb7 SQL, a full outer join must contain an equality comparison in the join condition.

## 3.6 Reserved Words and Keywords

In Rdb7 SQL, you can disable or enable keyword checking as defined by the ANSI/ISO SQL standard. If keyword checking is enabled, you cannot directly create or reference an object whose name is the same as a reserved word unless you use quotation marks (") around the object name or qualify the object name. (SQL\*Net for Rdb automatically sets ANSI quoting.)

For example:

```
SELECT CAST FROM TYPE;
      ^
%SQL-F-LOOK_FOR syntax error, looking for (, found FROM instead
SET ANSI QUOTING ON;
SELECT "CAST" FROM TYPE;
0 rows returned.
```

Appendix A contains a comprehensive table of reserved words and keywords.

To enable keyword checking in Rdb7 SQL, use the SET KEYWORD RULES statement. For more information regarding the SET KEYWORD RULES statement, see the *Oracle Rdb SQL Reference Manual*.

When using Oracle7 SQL, you get a different set of reserved words from those in Rdb7 SQL. One major difference is that Oracle7 SQL allows you to use reserved words that are built-in functions to name an object. For example, you can use DECODE as an object name without the required quotation marks or qualification that Rdb7 SQL does.

The following is a list of built-in function reserved words for Rdb7:

AVG  
CAST  
CHAR\_LENGTH  
CHARACTER\_LENGTH  
COALESCE  
CONCAT  
CONVERT  
COUNT  
DECODE  
EXTRACT  
LOWER  
MAX  
MIN  
NULLIF  
NVL  
OCTET\_LENGTH  
POSITION  
SUBSTRING  
SUM  
TRANSLATE  
TRIM  
UPPER

You can create an object using Rdb7 SQL using these built-in function reserved words. You are allowed to insert data into these objects or update these objects, however, you receive an informational message from Rdb7 SQL. You cannot, however, reference an object whose name is a built-in function reserved word unless you use ANSI quoting or qualify the object name.

The previous list of reserved words, which can sometimes be used for creating objects, but not for referencing them, is referred to as **partially reserved** for Rdb7 SQL.

---

---

# Query Specifications

This chapter compares Oracle7 and Rdb7 SQL pseudocolumns, built-in functions, functional operators, predicates, and select statements.

## 4.1 Pseudocolumns

The following sections provide information about specific aspects of pseudocolumns.

### 4.1.1 Row Addresses (ROWID and DBKEY)

#### **Common**

The Oracle7 SQL ROWID keyword is similar in format and function to the DBKEY and ROWID keywords for Rdb7 SQL. Each keyword returns a unique way of identifying a specific row of a table.

#### **Oracle7**

Application programs can query the row identification pseudocolumn using the ROWID keyword to obtain a unique way of referring to a specific row.

The ROWID keyword returns a value that is stored in a pseudocolumn of each underlying table that uniquely identifies each row. With Oracle7 SQL, the value of ROWID remains constant until the database is reorganized. The same value returned by the ROWID keyword may appear in different tables of a cluster.

Oracle7 SQL does not support the ROWID keyword for queries whose rows are not a subset of rows in an underlying table. Thus, a view whose query contains SELECT DISTINCT clauses, GROUP BY clauses, or aggregate functions, or is selected from more than one table, will not have a ROWID pseudocolumn.

**Rdb7**

Application programs can query the database key or row identification pseudocolumn using the DBKEY or ROWID keywords to obtain a unique way of referring to a specific row.

The ROWID keyword is a synonym to the DBKEY keyword. The ROWID and DBKEY keywords can refer to tables or views, but not to derived tables.

Rdb7 SQL supports concatenating database key or row identification pseudocolumns for views whose query specifications select rows from joined tables. Their values may not persist between sessions, or even between transactions, depending on how the user attached to the database (DBKEY SCOPE IS [ATTACH | TRANSACTION]).

## 4.1.2 Number of Rows Returned (ROWNUM)

**Common**

The ROWNUM keyword is used to restrict the number of rows returned in a query.

**Oracle7**

Consider the following information for the Oracle7 SQL ROWNUM keyword:

- You can use the ROWNUM keyword in a SELECT list.
- You can use the ROWNUM keyword in a subquery, rename the resulting pseudocolumn of values, and manipulate the pseudocolumn like any other column in the rest of the query.
- The ROWNUM keyword does not work with the ORDER BY clause. The ROWNUM keyword is applied first to limit the rows, then this random set is ordered.

**Rdb7**

Consider the following information for the Rdb7 SQL ROWNUM keyword:

- You can use the ROWNUM keyword only in a WHERE clause.
- The ROWNUM keyword works with the ORDER BY clause; that is, the resulting rows in the pseudocolumn are ordered before the number is limited.
- The ROWNUM keyword is similar to the LIMIT TO *n* ROWS clause.

### 4.1.3 Values of Sequences (NEXTVAL and CURRVAL)

**Common**

None.

**Oracle7**

With Oracle7 SQL, you can use the NEXTVAL and CURRVAL keywords when referring to values of sequences.

- The NEXTVAL keyword increments the sequence and returns the next value of the pseudocolumn.
- The CURRVAL keyword returns the current value of the sequence of the pseudocolumn.

You can use the NEXTVAL and CURRVAL keywords only in the VALUES clause of an INSERT statement or in the SET clause of an UPDATE statement.

Different users can use the same sequence simultaneously and each user receives different values for NEXTVAL.

**Rdb7**

The NEXTVAL and CURRVAL keywords are reserved for use in a future version of the Oracle Rdb database product.

### 4.1.4 Recursive Unions (LEVEL)

**Common**

None.

**Oracle7**

The LEVEL keyword is specific to Oracle7 SQL and returns values in a built-in pseudocolumn that is used in recursive unions.

**Rdb7**

Rdb7 SQL does not support the LEVEL keyword.

## 4.2 Functions

Table 4–1 contains a list of functions supported by the Oracle7 and Rdb7 SQL dialects. The asterisk ( \* ) indicates that the function is installed with SYSSLIBRARY:SQL\_FUNCTION[*nn*].SQL with Oracle Rdb7. The plus sign ( + ) indicates that the function is installed with SQL\*Net for Rdb.

**Table 4–1 Functions Supported by Oracle7 and Rdb7**

Function Name	Oracle7	Rdb7	Comments
ABS	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type. The Rdb7 SQL CASE expression can achieve the same results as the ABS function.
ACOS	Yes	No	
ADD_MONTHS	Yes	Yes	* Rdb7 SQL returns the DATE VMS data type.
ASCII	Yes	Yes	* Rdb7 SQL returns the INTEGER data type.
ASIN	Yes	No	
ATAN	Yes	No	
ATAN2	Yes	No	ATAN2 (x,y) = ATAN (x/y).
AVG	Yes	Yes	
CAST	No	Yes	
CEIL	Yes	Yes	* Ceil (n) returns the smallest integer >= n. Rdb7 SQL returns the DOUBLE PRECISION data type.
CHARACTER_LENGTH	No	Yes	CHARACTER_LENGTH is similar to the Oracle7 SQL LENGTH function.
CHARTOROWID	Yes	No	
CHR	Yes	Yes	* Converts the ASCII representation to the character. Rdb7 SQL returns the VARCHAR(2000) data type.
COALESCE	No	Yes	Similar to the NVL function.
CONCAT	Yes	Yes	Returns <i>char1</i> concatenated with <i>char2</i> .
CONVERT	Yes	Yes	
COS	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.

**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
COSH	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
COUNT	Yes	Yes	
CURRENT_DATE	No	Yes	
CURRENT_TIME	No	Yes	
CURRENT_TIMESTAMP	No	Yes	
CURRENT_USER	No	Yes	
DECODE	Yes	Yes	See Section 4.2.1 for more information.
DUMP	Yes	No	
EXP	Yes	Yes	* $e$ raised to the $n$ th power ( $e^{**n}$ ). Rdb7 SQL returns the DOUBLE PRECISION data type.
EXTRACT	No	Yes	
FLOOR	Yes	Yes	* Largest integer $\leq n$ . Rdb7 SQL returns the DOUBLE PRECISION data type.
GLB	Yes	No	
GREATEST	Yes	Yes	Rdb7 SQL only supports two arguments and returns the DOUBLE PRECISION data type.
GREATEST_LB	Yes	No	
HEXTORAW	Yes	Yes	* Rdb7 SQL returns the VARCHAR(2000) data type.
INITCAP	Yes	Yes	* INITCAP ( <i>text</i> ) converts the first letter of each word to uppercase. Rdb7 SQL returns the VARCHAR(2000) data type.

**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
INSTR	Yes	Yes	<p>* In Oracle7 SQL, the INSTR character function (<i>char-value</i>, <i>char-string</i> [,<i>n</i> [,<i>m</i>]]) returns the position of the <i>m</i>'th occurrence of <i>char-string</i> in <i>char-value</i> after the <i>n</i>'th character. The default values of <i>m</i> and <i>n</i> are 1. Note the following:</p> <ul style="list-style-type: none"> <li>■ If <i>char-value</i> or <i>char-string</i> is NULL, the result is NULL.</li> <li>■ If <i>char-string</i> does not appear, the result is 0.</li> <li>■ If <i>n</i> &lt; 0, the search is backwards from the end of <i>char-value</i>.</li> </ul> <p>INSTR is similar to the Rdb7 SQL POSITION character function. Rdb7 SQL returns the INTEGER data type.</p>
INSTRB	Yes	Yes	* Rdb7 SQL returns the INTEGER data type.
LAST_DAY	Yes	Yes	* Rdb7 SQL returns the DATE VMS data type.
LEAST	Yes	Yes	Rdb7 SQL only supports two arguments and returns the DOUBLE PRECISION data type.
LEAST_UB	Yes	No	
LENGTH	Yes	Yes	* LENGTH is similar to the Rdb7 SQL CHARACTER_LENGTH function. Rdb7 SQL returns the INTEGER data type.
LENGTHB	Yes	Yes	* LENGTHB is similar to the Rdb7 SQL OCTET_LENGTH function. Rdb7 SQL returns the INTEGER data type.
LN	Yes	Yes	* Natural log (base e). Rdb7 SQL returns the DOUBLE PRECISION data type.
LOG	Yes	Yes	* LOG ( <i>x</i> , <i>y</i> ) = LOG ( <i>y</i> ) (base <i>x</i> ). Rdb7 SQL returns the DOUBLE PRECISION data type.
LOWER	Yes	Yes	Returns <i>char</i> with all letters lowercase. LOWER is similar to the Oracle7 SQL NLS_LOWER function.



**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
LPAD	Yes	Yes	* LPAD ( <i>txt1</i> , <i>n</i> [, <i>txt2</i> ]) left pads <i>txt1</i> with <i>txt2</i> to length <i>n</i> . Rdb7 SQL returns the VARCHAR(2000) data type.
LTRIM	Yes	Yes	* LTRIM ( <i>txt1</i> [, <i>set</i> ]) removes characters in <i>set</i> from left of <i>txt1</i> . Rdb7 SQL returns the VARCHAR(2000) data type. LTRIM is similar to the Rdb7 SQL TRIM (LEADING ...) function.
LUB	Yes	No	
MAX	Yes	Yes	
MIN	Yes	Yes	
MOD	Yes	Yes	* MOD ( <i>x</i> , <i>y</i> ) = ( <i>x</i> - (FLOOR (ABS ( <i>x</i> / <i>y</i> )) * ABS ( <i>y</i> ))) SIGN ( <i>x</i> ). Rdb7 SQL returns the DOUBLE PRECISION data type.
MONTHS_BETWEEN	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
NEW_TIME	Yes	Yes	* Rdb7 SQL returns the DATE VMS data type.
NEXT_DAY	Yes	Yes	* Rdb7 SQL returns the DATE VMS data type.
NLS_INITCAP	Yes	No	NLS_INITCAP ( <i>txt1</i> [, ' <i>nlsparams</i> ']) national language set.
NLS_LOWER	Yes	No	Returns <i>char</i> with all letters in lowercase. NLS_LOWER is similar to the Rdb7 SQL LOWER function.
NLS_SORT	Yes	Yes	*
NLS_UPPER	Yes	No	Returns <i>char</i> with all letters in uppercase. NLS_UPPER is similar to the Rdb7 SQL UPPER function.
NULLIF	No	Yes	Can be simulated with the DECODE function.
NVL	Yes	Yes	NVL is similar to the Rdb7 SQL COALESCE expression.
OCTET_LENGTH	No	Yes	OCTET_LENGTH is similar to the Oracle7 SQL LENGTHB function.
POSITION	No	Yes	POSITION is similar to the Oracle7 SQL INSTR character function.

**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
POWER	Yes	Yes	* POWER ( $x,y$ ) = $x^{**}y$ . Rdb7 SQL returns the DOUBLE PRECISION data type.
RAWTOHEX	Yes	Yes	* Rdb7 SQL returns the VARCHAR(2000) data type.
REPLACE	Yes	Yes	* REPLACE ( <i>txt</i> , <i>search</i> , [ <i>replacement</i> ] <i>def</i> = “). Rdb7 SQL returns the VARCHAR(2000) data type.
ROUND	Yes	Yes	* ROUND ( $n$ [, $m$ ] default 0) = $n$ rounded to scale $m$ . Rdb7 SQL returns the DOUBLE PRECISION data type.
ROWIDTOCHAR	Yes	No	
RPAD	Yes	Yes	* Returns <i>char1</i> , right-padded to length $n$ with <i>char2</i> . Rdb7 SQL returns the VARCHAR(2000) data type.
RTRIM	Yes	Yes	* Returns <i>char</i> , with all the rightmost characters that appear in <i>set</i> removed. Rdb7 SQL returns the VARCHAR(2000) data type. RTRIM is similar to the Rdb7 SQL TRIM (TRAILING ...) function.
SESSION_USER	No	Yes	
SIGN	Yes	Yes	* SIGN ( $x$ ) = -1, 0, or 1. Rdb7 SQL returns the DOUBLE PRECISION data type. The Rdb7 SQL CASE expression can achieve the same results as the SIGN function.
SIN	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
SINH	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
SOUNDEX	Yes	No	Returns a character string containing the phonetic representation of <i>char</i> .
SQRT	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
STDDEV	Yes	No	

**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
SUBSTR	Yes	Yes	<p>* SUBSTR (<i>txt</i>, <i>m</i> [,<i>n</i>]) returns the substring of the source <i>txt</i>, as follows:</p> <ul style="list-style-type: none"> <li>■ If <math>m \geq 1</math>, the substring starts at position <i>m</i> where the first character is at position 1.</li> <li>■ If <math>m = 0</math>, -0, or 1, the substring starts as position 1.</li> <li>■ If <math>m &lt; 0</math>, the substring starts at the character <i>m</i> before the end of <i>txt</i> so it equals SUBSTRING (<i>txt</i> FROM LENGTH (<i>txt</i>) - ABS (<i>m</i>) [FOR <i>n</i>]).</li> </ul> <p>The Rdb7 SQL SUBSTRING function is similar to the Oracle7 SQL SUBSTR character string function. Rdb7 SQL returns the VARCHAR(2000) data type.</p>
SUBSTRB	Yes	Yes	<p>* SUBSTRB (<i>txt</i>, <i>m</i> [,<i>n</i>]) same but <i>m</i> and <i>n</i> in bytes not characters. Rdb7 SQL returns the VARCHAR(2000) data type.</p>
SUBSTRING	No	Yes	<p>SUBSTRING (<i>txt</i> FROM <i>m</i> [FOR <i>n</i>]) returns a substring of the source <i>txt</i>, starting at position <i>m</i>, where the first character in <i>txt</i> is at position 1.</p> <p>If <math>m \leq 0</math>, the substring starts counting from before the beginning of the <i>txt</i> until it reaches the beginning of <i>txt</i>, at position 1.</p> <p>The Oracle7 SQL SUBSTR function is similar to the Rdb7 SQL SUBSTRING function.</p>
SUM	Yes	Yes	<p>Oracle7 SQL does not issue an informational message when you select SUM (<i>x</i>) as is done with Rdb7 SQL.</p>
SYSDATE	Yes	Yes	<p>SYSDATE is similar to the Rdb7 SQL CURRENT_TIMESTAMP function. On distributed systems, the Oracle7 SQL SYS-DATE function returns the local date and time.</p>
SYSTEM_USER	No	Yes	

**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
TAN	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
TANH	Yes	Yes	* Rdb7 SQL returns the DOUBLE PRECISION data type.
TO_CHAR	Yes	Yes	+ Rdb7 SQL returns the VARCHAR(2000) data type.
TO_DATE	Yes	Yes	+ Rdb7 SQL returns the DATE VMS data type.
TO_LABEL	Yes	No	
TO_MULTI_BYTE	Yes	No	
TO_NUMBER	Yes	Yes	+ Rdb7 SQL returns the DOUBLE PRECISION data type.
TO_SINGLE_BYTE	Yes	No	
TRANSLATE	Yes	Yes	For Oracle7 SQL, TRANSLATE ( <i>txt, fromset, toset</i> ) replaces each <i>char</i> by the corresponding <i>char</i> in <i>toset</i> or blank.  For Rdb7 SQL, TRANSLATE ( <i>char USING translation-name</i> ) translates a character value expression from one character set to another compatible character set.
TRIM	No	Yes	TRIM (LEADING ...) is similar to the Oracle7 SQL LTRIM function. TRIM (TRAILING ...) is similar to the Oracle7 SQL RTRIM function.
TRUNC	Yes	Yes	* ROUND ( <i>n [,m]</i> default 0) = <i>n</i> truncated to scale <i>m</i> . Rdb7 SQL returns the DOUBLE PRECISION data type.
UID	Yes	No	
UPPER	Yes	Yes	Returns <i>char</i> with all letters in uppercase. UPPER is similar to the Oracle7 SQL NLS_UPPER function.

**Table 4–1 Functions Supported by Oracle7 and Rdb7(Cont.)**

Function Name	Oracle7	Rdb7	Comments
USER	Yes	Yes	USER is similar for both Oracle7 and Rdb7 SQL, with the following differences in data types: <ul style="list-style-type: none"> <li>▪ The Oracle7 SQL data type is VARCHAR2(30).</li> <li>▪ The Rdb7 SQL data type is VARCHAR(31).</li> </ul>
USERENV	Yes	Yes	+ USERENV (session option) returns the VARCHAR2 data type. Rdb7 SQL returns the VARCHAR(2000) data type.
VARIANCE	Yes	No	
VSIZE	Yes	No	

\* The asterisk indicates that the function is installed with SYSSLIBRARY:SQL\_FUNCTIONS[*nn*].SQL with Oracle Rdb7.

+ The plus sign indicates that the function is installed with SQL\*Net for Rdb.

## 4.2.1 DECODE Function

### Common

Both Oracle7 and Rdb7 SQL support the DECODE operator, which translates *source* into the *translate-expression* corresponding to the first matching *match-expression*. It is similar in function to the Rdb7 SQL CASE operator.

### Oracle7

None of the arguments can contain subqueries. The arguments may, however, include the ROWID and ROWNUM keywords.

In Oracle7 SQL, an expression evaluating to NULL will match another expression evaluating to NULL.

### Rdb7

The arguments may contain subqueries. The arguments may include the ROWID keyword, however, may not include the ROWNUM keyword.

---



---

**Note:** If source evaluates to NULL, then the DECODE operator will match a *match-expression*, which is the literal NULL, but not an expression that evaluates to NULL, thus:

```
Rdb_SQL> SELECT * FROM t1;
          X      Y
          1      one
          NULL    two

Rdb_SQL> SELECT DECODE (x, null, 'same', 'other') FROM t1;
          other
          same
2 rows selected

Rdb_SQL> SELECT DECODE (null, x, 'same', 'other') FROM t1;
          other
          other
2 rows selected
```

---



---

## 4.2.2 CASE Expression

### Common

None.

### Oracle7

Oracle7 SQL does not support the CASE expression.

### Rdb7

The CASE expression is considered a generalization of the Oracle7 SQL DECODE function.

## 4.3 Predicates

### Common

Both Oracle7 and Rdb7 SQL support the ANY, ALL, EXISTS, IN, IS NULL, and SOME predicates.

The LIKE predicate is supported by Oracle7 and Rdb7 SQL, with the following differences:

- LIKE with numeric comparison

- Oracle7 SQL allows you to use the LIKE predicate with numeric data.
- Rdb7 SQL requires explicit casting of numeric data to a character data type.
- LIKE with a select subquery
  - Oracle7 SQL allows a variable or column name, but does not allow a subquery.
  - Rdb7 SQL allows the pattern to be any expression of character data type.
- Case sensitivity
  - Oracle7 SQL matches the case when making a comparison. To override this, so that the column or variable *x* containing the value 'aBCd' is LIKE 'Ab%', translate both terms to uppercase as in UPPER (*x*) LIKE 'AB%'.
  - Rdb7 SQL by default matches the case. To override the case, use the IGNORE CASE clause.

Although Oracle7 and Rdb7 SQL differ in how they compare varying-length character fields, they both recognize trailing blanks in VARCHAR fields when evaluating the LIKE predicate.

### Oracle7

Comparisons between rows are supported by Oracle7 SQL. For example:

```
SELECT * FROM employees
WHERE   (last_name, first_name, middle_initial)
        IN (SELECT last_name, first_name, middle_initial
            FROM   stock_options ...
            );
```

Oracle7 SQL follows the ANSI/ISO SQL standard.

### Rdb7

The following predicates are also supported by Rdb7 SQL:

- CONTAINING
- SINGLE
- STARTING WITH

Rdb7 SQL allows comparisons between single elements only.

## 4.4 Select Expression

### Common

Both Oracle7 and Rdb7 SQL support the UNION and UNION ALL set operators in query expressions to combine the rows returned by two SELECT statements.

### Oracle7

Consider the following information for the Oracle7 SQL select expression:

- Recursive unions are supported by Oracle7 SQL.  
Oracle7 SQL supports a CONNECT BY clause in the SELECT statement that specifies a hierarchical relationship between parent rows and child rows.  
This is a subset of the SQL3 recursive union capability using completely different syntax.
- Oracle7 SQL supports the MINUS set operator and the ANSI/ISO SQL standard INTERSECT set operator. The MINUS set operator is equivalent to the ANSI/ISO SQL standard EXCEPT set operator.
- Oracle7 SQL does not support the LIMIT TO clause. However, you can achieve similar results with the ROWNUM keyword (see Section 4.1.2). For example:

```
... WHERE ROWNUM ≤ 10
```

Oracle7 SQL does not support the OPTIMIZE FOR clause. However, you can achieve similar results using the /\*+ ... +\*/ syntax. See the Oracle7 documentation for more information.

- Oracle7 SQL has a maximum size that can be returned by a component of a UNION, MINUS, or EXCEPT, which is determined by the value of the initialization parameter DB\_BLOCK\_SIZE.

### Rdb7

Consider the following information for the Rdb7 SQL select expression:

- LIMIT TO and ROWNUM keywords  
Rdb7 SQL limits the number of rows returned by a select expression to be at most the number that you specify in a LIMIT TO clause. For example:

```
SELECT last_name FROM employees ORDER BY salary LIMIT TO 10 ROWS;
```

See Section 4.1.2 for additional information.



- **OPTIMIZE FOR** clause  
Rdb7 SQL allows you to indicate the preferred optimizer strategy. For example:  

```
SELECT last_name FROM employees WHERE... OPTIMIZE FOR FAST FIRST;
```
- Rdb7 does not support recursive unions.
- Rdb7 does not support the MINUS and INTERSECT set operators.

## 4.4.1 SELECT List

### Common

In a SELECT list, both Oracle7 and Rdb7 SQL permit the asterisk (\*) to be qualified. For example:

```
SELECT t1.* FROM t1, t2 WHERE t1.x = t2.x;
```

### Oracle7

Only Oracle7 SQL allows you to use the following syntax that is an extension to the ANSI/ISO SQL standard:

```
SELECT 1 FROM t1, t1, t1
```

You cannot select anything other than a literal from such a multijoin without providing correlation names.

Oracle7 SQL supports, as obsolete syntax, the SELECT... UNIQUE statement where UNIQUE is a synonym for DISTINCT.

### Rdb7

In a select list, Rdb7 SQL allows you to list explicit columns and expand the statement. For example:

```
SELECT ROWID, * FROM t1;
```

Rdb7 SQL supports subqueries as members of a select list.

## 4.4.2 Derived Tables

### Common

Both Oracle7 and Rdb7 SQL support using derived tables and renaming columns, however, the syntax is slightly different.

**Oracle7**

For renaming columns and setting derived tables, Oracle7 SQL uses the following syntax:

```
(SELECT x [[AS] external-col-name] FROM t...) [correlation-name]
```

Thus, in Oracle7 SQL, the columns are renamed (for reference outside the query specification) in the select list, with an optional AS keyword. The keyword AS cannot be given before the *correlation-name* of the derived table.

Also, the *correlation-name* is optional unless a reference to a column needs to be qualified.

**Rdb7**

Rdb7 SQL supports the ANSI/ISO SQL standard syntax when renaming columns and setting derived tables, as shown in the following example:

```
(SELECT x [[AS] renamed-column] FROM t...)
[AS] correlation-name [(renamed-column-list)]
```

Thus, in Rdb7 SQL, the columns can be renamed inside the list, or outside with the *correlation-name* for the derived table. Any *renamed-column-list* will supersede any *renamed-column* given in the select list.

Rdb7 SQL requires that a *correlation-name* be given for the derived table.

### 4.4.3 Scope of Table Names

**Common**

According to the ANSI/ISO SQL standard, the scope of a table name (or of its correlation name if one is provided) specified in the FROM clause of a query includes the SELECT list and the entire WHERE clause, but excludes any derived tables in the same FROM clause. Consider the following example:

```
SELECT select-list FROM t1,
      (SELECT select-list2 FROM t2 WHERE t2.x = t1.y)
tmp WHERE...
```

The reference inside the derived table *tmp* to *t1.y* is invalid.

**Oracle7**

Oracle7 SQL returns an error in the example cited in the Common subheading of this section.

**Rdb7**

Rdb7 SQL allows the syntax cited in the example in the Common subheading of this section when the table is specified before the reference in the derived table. If the order is reversed, Rdb7 SQL returns an error. For example:

```
SELECT select-list FROM
    (SELECT select-list2 FROM t2 WHERE t2.x = t1.y)
tmp t1 WHERE...;
```

#### 4.4.4 ORDER BY Clause

**Common**

In both Oracle7 and Rdb7 SQL, the sort items in an ORDER BY clause can be column names or integer literals, denoting the ordinal position of the item in the select list on which to sort.

**Oracle7**

Also supports ordering by an expression. For example:

```
SELECT to_char (hiredate, 'YY') FROM emp
GROUP BY to_char (hiredate, 'YY')
ORDER BY to_char (hiredate, 'YY');
```

Oracle7 SQL does not support ORDER BY in subqueries and views.

**Rdb7**

Rdb7 SQL does not allow ordering by an expression. To work around this, you can give the calculated expression a correlation name, which entails putting it in the select list. For example:

```
SELECT to_char (hiredate, 'YY') AS year FROM emp
GROUP BY year
ORDER BY year;
```

Or, so as not to have to select the expression, calculate it in a derived table. For example:

```
SELECT x FROM (SELECT x, NVL (y, 'n', y) AS n FROM t1)
AS calc
ORDER BY n;
```

Rdb7 SQL supports ORDER BY in subqueries and views.

## 4.4.5 GROUP BY Clause

### Common

Both Oracle7 and Rdb7 SQL support the GROUP BY clause.

### Oracle7

The GROUP BY and HAVING clauses can appear in any order in a query. For example:

```
SELECT to_char (hiredate, 'YY') FROM emp
HAVING year > '90'
GROUP BY to_char (hiredate, 'YY')
ORDER BY to_char (hiredate, 'YY');
```

### Rdb7

The GROUP BY clause must appear before the HAVING clause in a query. For example:

```
SELECT to_char (hiredate, 'YY') AS year FROM emp
GROUP BY year
HAVING year > '90'
ORDER BY year;
```

---

---

# Data Manipulation

This chapter compares the Oracle7 and Rdb7 SQL statements for database manipulation, including database attachments and data updates, deletes, and inserts.

## 5.1 Database Attachments

### Common

None.

### Oracle7

Oracle7 SQL allows access to tables in other databases through database links. For example:

```
... FROM schema.table@db_link
```

### Rdb7

Rdb7 SQL allows a single statement to refer to more than one database attachment only in a cross-database INSERT statement. For example:

```
INSERT INTO db1.table1(col1) SELECT col2 FROM db2.table2;
```

The cross-database INSERT statement is not supported in either a stored routine or in a multistatement procedure.

## 5.2 Subqueries (UPDATE, DELETE and INSERT)

### Common

Although the syntax for the Oracle7 SQL usage of the UPDATE, DELETE, and INSERT statements is similar to the Rdb7 SQL usage, Oracle7 SQL allows you to supply a subquery while Rdb7 SQL allows you to specify only a table name.

### Oracle7

For the following statements, Oracle7 SQL allows you to supply a subquery, a table name, or a view name:

- DELETE [FROM] *subquery* or *table-name* or *view-name*
- INSERT INTO *subquery* or *table-name* or *view-name*
- UPDATE *subquery* or *table-name* or *view-name*

The subquery or view must be updatable.

This is an extension to the ANSI/ISO SQL standard.

### Rdb7

For the following statements, Rdb7 SQL requires that you supply a table or view name:

- DELETE [FROM] *table-name* or *view-name*
- INSERT INTO *table-name* or *view-name*
- UPDATE *table-name* or *view-name*

## 5.3 UPDATE Statement

The following subsections compare specific aspects of the UPDATE statement for the Oracle7 and Rdb7 SQL dialects.

### 5.3.1 Set Single or Multiple Columns (SET Clause)

#### Common

Both Oracle7 and Rdb7 SQL support the SET clause.

**Oracle7**

Oracle7 SQL allows multiple columns to be set in a single SET clause assignment to the result of a subquery. For example:

```
UPDATE employees
   SET   (salary, bonus) = SELECT (value1, value2) FROM t1 ...
  WHERE ...
```

**Rdb7**

Rdb7 SQL supports only one column per SET clause.

### 5.3.2 Mix Values and Subqueries (UPDATE and INSERT Statements)

**Common**

Both Oracle7 and Rdb7 SQL support mixing values and subqueries.

**Oracle7**

Oracle7 SQL allows you to either insert or update the results of a subquery directly, or list the values separately. However, you cannot list values and subqueries in the same clause. For example, notice the error messages returned if you try to mix listing values and subqueries in the same UPDATE or INSERT statement:

```
UPDATE t1 SET (x, y) = (13, (SELECT c2 FROM t2 WHERE c1 = 'eins'));
```

ERROR at line 2:

ORA-01767: UPDATE ... SET expression must be a subquery

```
INSERT INTO t2 (c2, c1) VALUES (13, (SELECT y FROM t1 WHERE x = 13));
```

```
INSERT INTO t2 (c2, c1) VALUES (13, (SELECT y FROM t1 WHERE x = 13))
```

\*

ERROR at line 1:

ORA-00936: missing expression

To perform this UPDATE or INSERT, put the constant value into the subquery. For example:

```
INSERT INTO t2 (c2, c1) SELECT 13, y FROM t1 WHERE x = 13;
```

Oracle7 SQL does not allow you to update the same column more than once in the same UPDATE statement.

**Rdb7**

Rdb7 SQL allows you to list both values and subqueries in INSERT statements. For example, Rdb7 SQL supports the following INSERT statement:

```
INSERT INTO t2 (c2, c1) VALUES (13, (SELECT y FROM t1 WHERE x = 13));
1 row inserted
```

Rdb7 SQL allows you to update the same column more than once in the same UPDATE statements.

### 5.3.3 Bulk Updates on a Column with a UNIQUE Constraint

**Common**

Bulk updates are handled similarly by both the Oracle7 and the Rdb7 SQL UPDATE statements.

**Oracle7**

Oracle7 SQL handles bulk updates on a UNIQUE column without problem.

**Rdb7**

Rdb7 SQL has a limitation when the column to be updated has a UNIQUE constraint. If the intermediate results (when some but not all the rows have been updated) are not unique:

- Rdb7 SQL fails if the dialect has not been set to Oracle Level1 before attaching to the database. (The Oracle Level1 dialect is the default for SQL\*Net for Rdb.)
- Rdb7 SQL fails if there is a UNIQUE INDEX on the column in question. You may be able to work around this by creating a view, ordering the column in reverse order, and updating the view. For example:

```
SELECT * FROM t1;
X      Y
1      eins
2      zwei
UPDATE t1 SET x = x + 1;
%RDB-E-NO_DUP, index field value already exists duplicate values not allowed to t1
CREATE VIEW vt1 AS SELECT * FROM t1 ORDER BY x DESC;
UPDATE vt1 SET x = x + 1;
2 rows updated
SELECT * FROM t1;
X      Y
2      eins
3      zwei
```



```
2 rows selected
DROP VIEW vt1;
```

### 5.3.4 Correlated Subqueries

#### Common

Although the UPDATE statement syntax is similar for both Oracle7 and Rdb7 SQL, correlated subqueries are handled in a slightly different way as described in the following subheadings.

#### Oracle7

If the replacement value has a subquery correlated to the table being updated, then Oracle7 SQL evaluates the subquery once for each row. For example:

```
UPDATE t1
SET x = (SELECT t2.x1 from t2 where t2.x = t1.y),
      y = (SELECT t2.y1 from t2 where t2.y = t1.x);
```

#### Rdb7

Rdb7 SQL follows the ANSI/ISO SQL standard, which specifies that all subqueries should be evaluated ahead of time.

## 5.4 DELETE [FROM] Table

#### Common

Oracle7 and Rdb7 SQL both support the basic DELETE feature and syntax.

#### Oracle7

The keyword FROM is optional in an Oracle7 SQL DELETE statement. This is an extension to the ANSI/ISO SQL standard.

#### Rdb7

Rdb7 SQL requires the FROM keyword.

DELETE [FROM] Table

---

---

---

# Data Definition Statements

This chapter compares Oracle7 and Rdb7 SQL statements used for data definition.

## 6.1 Data Definition Language and Transactions

### Common

When you use SQL\*Net for Rdb, it performs an implicit COMMIT statement before and after executing a DDL statement. This is the same action taken by Oracle7 SQL.

### Oracle7

In Oracle7 SQL, a data definition language (DDL) statement cannot be rolled back. This is because Oracle7 SQL implicitly commits data before and after executing a DDL statement.

### Rdb7

In Rdb7 SQL, a DDL statement does not perform an implicit COMMIT statement. Consequently, Rdb7 SQL allows you to roll back a DDL statement.

## 6.2 Data Type Usage (CREATE TABLE, CREATE DOMAIN, Declare Variable)

The following sections describe data type synonyms and defaults for Oracle7 and Rdb7 SQL as used when creating and altering columns of a table, creating domains (in Rdb7 SQL), or declaring variables or parameters.

## 6.2.1 Data Type Synonyms

### Common

None.

### Oracle7

The following list describes data type synonyms and equivalents for Oracle7 SQL:

- CHARACTER and CHAR are synonyms.
- CHAR VARYING, CHARACTER VARYING, and CHARVAR2 are synonyms.
- NUMERIC (p), DECIMAL (p), and DEC (p) are synonyms and are equivalent to NUMBER (p, 0).
- INTEGER, INT, and SMALLINT are synonyms and are equivalent to NUMBER (38).
- DOUBLE PRECISION and FLOAT are equivalent to FLOAT (126).
- REAL is equivalent to FLOAT (63).
- LONG and LONG VARCHAR are synonyms.

### Rdb7

The following list describes data type synonyms and equivalents for Rdb7 SQL:

- CHARACTER and CHAR are synonyms.
- NCHAR, NATIONAL CHAR, and NATIONAL CHARACTER are synonyms.
- NCHAR VARYING, NATIONAL CHAR VARYING, and NATIONAL CHARACTER VARYING are synonyms.
- INTEGER and INT are synonyms.
- LONG VARCHAR is equivalent to VARCHAR (16383).

## 6.2.2 Data Type Defaults

Table 6–1 describes data type defaults for Oracle7 and Rdb7 SQL.

**Table 6–1 Data Type Defaults for Oracle7 and Rdb7 SQL**

Data Type	Oracle7 Default	Rdb7 Default
CHAR (n)	Default value for $n = 1$	Default value for $n = 1$

**Table 6–1 Data Type Defaults for Oracle7 and Rdb7 SQL (Cont.)**

<b>Data Type</b>	<b>Oracle7 Default</b>	<b>Rdb7 Default</b>
VARCHAR ( <i>n</i> ) VARCHAR2( <i>n</i> )	No default value	No default value
Numeric data type, scale <i>s</i> , precision <i>p</i>	Default for <i>s</i> = 0 Default for <i>p</i> = 38	Default for <i>s</i> = 0 Default for <i>p</i> = 5
FLOAT ( <i>p</i> )	Default value for <i>p</i> = 126	Default value for <i>p</i> = 53
DATE	Default date format is set by initialization parameter NLS_DATE_FORMAT or NLS_TERRITORY	DATE VMS or DATE ANSI format is set by the value of DEFAULT DATE FORMAT
INTERVAL ( <i>f</i> )	Does not apply	Default value for <i>f</i> = 2
TIMESTAMP ( <i>f</i> )	Does not apply	Default value for <i>f</i> = 2
TIME ( <i>f</i> )	Does not apply	Default value for <i>f</i> = 0
LIST OF BYTE VARYING ( <i>n</i> )	Does not apply	Default value for <i>n</i> = 1  If <i>n</i> is specified as 0, Rdb7 SQL interprets this as 512.

## 6.3 Table Creation (CREATE TABLE ... AS SELECT)

### Common

Both Oracle7 and Rdb7 SQL support the CREATE TABLE statement as defined by the ANSI/ISO SQL standard.

### Oracle7

Oracle7 SQL supports an optional AS subquery to specify the rows that should be inserted into the table upon its creation. For example, the following SQL statement includes the AS subquery:

```
CREATE TABLE sales (badge CHAR (5), quota DECIMAL (9)
AS (SELECT badge, NULL FROM employees WHERE job = 'Sales');
```

This statement is performed as an atomic unit. Therefore, if the transaction is unable to insert the rows returned by the AS subquery, the table is not created.

### **Rdb7**

Rdb7 SQL supports creating a table and inserting data as two separate statements.

## **6.4 Table Alterations (ALTER TABLE)**

### **Common**

Both Oracle7 SQL and Rdb7 SQL allow alterations to a table (for example, adding a column or changing a column data type).

### **Oracle7**

Oracle7 SQL ALTER TABLE statement allows you to disable or enable a constraint or all triggers.

Oracle7 SQL does not allow you to delete a column from a table.

### **Rdb7**

Rdb7 SQL supports adding computed columns, which can refer to other tables. For example:

```
ALTER TABLE t1 (ADD COLUMN c COMPUTED BY
    (SELECT MAX (y) FROM t2 WHERE x = a));
```

## **6.5 Constraints**

The following sections provide information regarding constraints.

### **6.5.1 Referential Delete (ON DELETE CASCADE)**

#### **Common**

None.

#### **Oracle7**

The referential delete capability is unique to Oracle7 SQL. Oracle7 SQL allows you to specify the ON DELETE CASCADE clause when defining a referential constraint.

This complies with the ANSI/ISO SQL standard.

#### **Rdb7**

The referential delete capability is not supported by Rdb7 SQL.

## 6.5.2 Unique Indexes (UNIQUE or PRIMARY KEY)

### Common

Both Oracle7 and Rdb7 SQL support UNIQUE and PRIMARY KEY indexes.

### Oracle7

Oracle7 SQL implicitly creates an index to enforce a UNIQUE or PRIMARY KEY constraint whose parameters can be set inside the CREATE TABLE statement.

### Rdb7

Rdb7 SQL does not create a unique index without an explicit CREATE INDEX statement.

## 6.5.3 Constraint Exceptions (EXCEPTIONS INTO)

### Common

None.

### Oracle7

The Oracle7 SQL EXCEPTIONS INTO *table-name* clause (used in the CREATE TABLE or ALTER TABLE statements) specifies a table into which information about table rows that do not meet a defined constraint will be stored.

### Rdb7

The constraint exceptions function is not supported by Rdb7 SQL.

## 6.5.4 Disabled and Enabled Constraints (DISABLE, ENABLE)

### Common

None.

### Oracle7

Oracle7 SQL allows individual constraints to be enabled and disabled. Each constraint can, optionally, be disabled when the table is created. Another option is to alter the table to enable or disable constraints. For example:

```
ALTER TABLE employees DISABLE PRIMARY KEY;  
ALTER TABLE employees ENABLE CONSTRAINT check_dependents;
```

### **Rdb7**

To disable a constraint, delete and redefine it.

## 6.5.5 Deferred Constraints

### **Common**

None.

### **Oracle7**

In Oracle7 SQL, all constraints can be deferred in that they can be disabled and enabled again at commit time.

### **Rdb7**

Rdb7 SQL allows constraints to be defined as deferrable or not deferrable.

A constraint can be DEFERRED or IMMEDIATE as part of a SET TRANSACTION or DECLARE TRANSACTION statement. You can also specify the SET ALL CONSTRAINTS IMMEDIATE or SET ALL CONSTRAINTS DEFERRED statement. Also see the SET TRANSACTION ... EVALUATING clause in the *Oracle Rdb7 SQL Reference Manual*.

## 6.6 CREATE OR REPLACE Clause for Compiled Objects

### **Common**

None.

### **Oracle7**

Oracle7 SQL supports creating or re-creating compiled objects, which may or may not exist, using the CREATE OR REPLACE clause.

The compiled objects can be a function, package, package body, procedure, trigger, or view. If the object does not exist, it is created. If the object does exist, it is replaced. There is no need to revoke and grant privileges again on the object.

### **Rdb7**

In Rdb7 SQL, you must first delete the object, if it exists, then define it again.



## 6.7 CREATE VIEW Statement

### Common

Both Oracle7 and Rdb7 SQL support the ANSI/ISO SQL standard CREATE VIEW statement.

### Oracle7

Oracle7 SQL also supports the following:

- CREATE OR REPLACE VIEW  
See Section 6.6 for more information.
- FORCE or NOFORCE  
FORCE allows you to create a view whose underlying tables have not yet been created. NOFORCE requires the underlying tables exist and be accessible to the user. NOFORCE is the default.
- READ ONLY  
Specifies that the view cannot be used for updates.

### Rdb7

Rdb7 SQL supports the ANSI/ISO SQL standard implementation of the CREATE VIEW statement.

## 6.8 Add Comments to Describe Metadata or Data Definitions (COMMENT ON)

### Common

Both Oracle7 and Rdb7 SQL support storing comments about a table or column definition in the database.

### Oracle7

Oracle7 SQL also supports comments on views and snapshots.

### Rdb7

Consider the following for the Oracle Rdb7 COMMENT ON statement:

- Rdb7 SQL supports a COMMENT ON statement to create a comment for catalogs, schemas, tables, domains, indexes, and columns.
- Comments for collating sequences, stored modules, outlines, and stored routines are part of their definitions and cannot be changed.
- You cannot supply a comment for a view definition.

## 6.9 Delete Definitions (DROP ... CASCADE)

### Common

The DROP statement is similar in both Oracle7 and Rdb7 SQL if the object being deleted (dropped) has no other objects on which it depends.

### Oracle7

Oracle7 SQL does not support the ANSI/ISO SQL standard semantics of the DROP statement and does not support the CASCADE and RESTRICT keywords. Instead, when you delete an object, (DROP\_O) such as a table, Oracle7 SQL marks the dependent objects (such as views that refer to it) as invalid.

Subsequently, if you later try to reference a dependent object DEP\_O, the system attempts to recompile it as described in the following table:

If . . .	Then . . .
DROP_O has been re-created	The recompile operation might be successful.
DROP_O has not been re-created	The user is notified that DEP_O is invalid.

### Rdb7

Rdb7 SQL supports the CASCADE and RESTRICT keywords in compliance with the ANSI/ISO SQL standard. To delete an object on which other objects depend, you must specify DROP ... CASCADE. In this case, the dependent object is also, implicitly, deleted or marked as invalid in CASCADE mode.

## 6.10 Object Recompilation (ALTER FUNCTION, ALTER TRIGGER, ALTER VIEW, ALTER PROCEDURE)

### Common

None.

**Oracle7**

Oracle7 SQL supports the ALTER FUNCTION, ALTER PROCEDURE, ALTER TRIGGER, and ALTER VIEW statements to recompile invalid functions, procedures, triggers, and views, respectively. These objects become invalid, for instance, when an underlying table is deleted.

**Rdb7**

Rdb7 SQL does not support the capabilities described for Oracle7 SQL in this section.

## 6.11 Rename Database Objects (RENAME)

**Common**

None.

**Oracle7**

Oracle7 SQL allows a table, view, sequence, or private synonym to be renamed. For example:

```
RENAME employees TO local_employees;
```

The RENAME statement is considered to be a DDL statement. Thus, the data is implicitly committed immediately before and after executing a DDL statement. (See Section 6.1.)

**Rdb7**

Rdb7 SQL does not support renaming database objects.



---



---

## Reserved Words

The following table lists the known reserved words as defined by the ANSI/ISO SQL standard. The second column of the table lists the specific ANSI standard in which the word has been reserved. The last three columns of the table show the differences in semantics between the Oracle Rdb Oracle Level1 dialect, the Oracle7 dialect, and the Oracle Rdb Oracle Level2 dialect (when implemented) when keyword checking is disabled. For example, ADD was reserved in the SQL92 ANSI standard; it is a keyword for the Oracle Rdb Oracle Level1 dialect, a reserved word for the Oracle7 dialect, and will be a reserved word for the Oracle Rdb Oracle Level2 dialect.

**Table A-1 Comparison of Reserved Words and Semantics**

Reserved Word	ANSI/ISO SQL Standard	Oracle Rdb Oracle Level1 Dialect	Oracle7	Oracle Rdb Oracle Level2 <sup>1</sup> Dialect
ABSOLUTE	SQL92	Keyword		Keyword
ACCESS				Reserved
ACTION	SQL92	Keyword		Keyword
ADD	SQL92	Keyword	Reserved	Reserved
ADMIN	SQL92	Keyword	Keyword	Keyword
AFTER	SQL3	Keyword	Keyword	Keyword
ALIAS	SQL3	Keyword		Keyword
ALL	SQL89	Partially Reserved	Reserved	Reserved
ALLOCATE	SQL92	Keyword	Keyword	Keyword
ALTER	SQL92	Keyword	Reserved	Reserved
AND	SQL89	Keyword	Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
ANY	SQL89	Keyword	Reserved	Reserved
ARE	SQL92	Keyword		Keyword
AS	SQL89	Partially Reserved	Reserved	Reserved
ASC	SQL89	Keyword	Reserved	Reserved
ASSERTION	SQL92	Keyword		Keyword
AT	SQL92	Keyword	PL/SQL	Keyword
AUDIT			Reserved	Reserved
AUTHORIZATION	SQL89	Keyword	Keyword	Keyword
AVG	SQL89	Partially Reserved	PL/SQL	Keyword
BEFORE	SQL3	Keyword	Keyword	Keyword
BEGIN	SQL89	Keyword	Keyword	Keyword
BETWEEN	SQL89	Keyword	Reserved	Reserved
BIT	SQL92	Keyword		Keyword
BIT_LENGTH	SQL92	Keyword		Keyword
BOOLEAN	SQL3	Keyword	PL/SQL	Keyword
BOTH	SQL92	Keyword		Keyword
BREADTH	SQL3	Keyword		Keyword
BY	SQL89	Keyword	Reserved	Reserved
CASCADE	SQL92	Keyword	Keyword	Keyword
CASCADEDED	SQL92	Keyword		Keyword
CASE	SQL92	Partially Reserved		Reserved
CAST	SQL92	Partially Reserved		Keyword
CATALOG	SQL92	Keyword		Keyword
CHAR_LENGTH	SQL92	Partially Reserved		Keyword
CHAR	SQL89	Keyword	Reserved	Reserved
CHARACTER_LENGTH	SQL92	Partially Reserved		Keyword
CHARACTER	SQL89	Keyword	Keyword	Keyword
CHECK	SQL89	Keyword	Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
CLASS	SQL3	Keyword		Keyword
CLOSE	SQL89	Keyword		Keyword
CLUSTER			Reserved	Reserved
COALESCE	SQL92	Partially Reserved		Keyword
COBOL	SQL89	Keyword	Keyword	Keyword
COLLATE	SQL92	Keyword		Keyword
COLLATION	SQL92	Keyword		Keyword
COLUMN	SQL92	Keyword	Reserved	Reserved
COMMENT			Reserved	Reserved
COMMIT	SQL89	Keyword	PL/SQL	Keyword
COMPLETION	SQL3	Keyword		Keyword
COMPRESS			Reserved	Reserved
COMPUTED		Partially Reserved		Reserved
CONCAT		Partially Reserved		Keyword
CONNECT	SQL92	Keyword	Reserved	Reserved
CONNECTION	SQL92	Keyword		Keyword
CONSTRAINT	SQL92	Keyword	Keyword	Keyword
CONSTRAINTS	SQL92	Keyword	Keyword	Keyword
CONSTRUCTOR	SQL3	Keyword		Keyword
CONTINUE	SQL89	Keyword	Keyword	Keyword
CONVERT	SQL92	Partially Reserved		Keyword
CORRESPONDING	SQL92	Keyword		Keyword
COUNT	SQL89	Partially Reserved	Keyword	Keyword
CREATE	SQL89	Keyword	Reserved	Reserved
CROSS	SQL92	Keyword		Keyword
CURRENT_DATE	SQL92	Partially Reserved		Reserved
CURRENT_TIME	SQL92	Partially Reserved		Reserved
CURRENT_TIMESTAMP	SQL92	Partially Reserved		Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
CURRENT_USER	SQL92	Partially Reserved		Reserved
CURRENT	SQL89	Partially Reserved	Reserved	Reserved
CURSOR	SQL89	Keyword	PL/SQL	Keyword
CYCLE	SQL3	Keyword	Keyword	Keyword
DATA	SQL3	Keyword		Keyword
DATE	SQL92	Partially Reserved	Reserved	Reserved
DAY	SQL92	Keyword		Keyword
DBHIGH				Reserved
DBKEY		Partially Reserved		Reserved
DBLOW				Reserved
DBMAC				Reserved
DEALLOCATE	SQL92	Keyword		Keyword
DEC	SQL89	Keyword	Keyword	Keyword
DECIMAL	SQL89	Keyword	Reserved	Reserved
DECLARE	SQL89	Keyword	PL/SQL	Keyword
DECODE		Partially Reserved		
DEFAULT	SQL89	Keyword	Reserved	Reserved
DEFERRABLE	SQL92	Keyword		Keyword
DEFERRED	SQL92	Keyword		Keyword
DELETE	SQL89	Keyword	Reserved	Reserved
DEPTH	SQL3	Keyword		Keyword
DEREF	SQL3	Keyword		Keyword
DESC	SQL89	Keyword	Reserved	Reserved
DESCRIBE	SQL92	Keyword		Keyword
DESCRIPTOR	SQL92	Keyword		Keyword
DESTROY	SQL3	Keyword		Keyword
DESTRUCTOR	SQL3	Keyword		Keyword
DIAGNOSTICS	SQL92	Keyword		Keyword



**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
DICTIONARY	SQL3	Keyword		Keyword
DISCONNECT	SQL92	Keyword		Keyword
DISTINCT	SQL89	Partially Reserved	Reserved	Reserved
DO	SQL3	Keyword	PL/SQL	Keyword
DOMAIN	SQL92	Keyword		Keyword
DOUBLE	SQL89	Keyword	Keyword	Keyword
DROP	SQL92	Keyword	Reserved	Reserved
EACH	SQL3	Keyword	Keyword	Keyword
ELEMENT	SQL3	Keyword		Keyword
ELSE	SQL92	Keyword	Reserved	Reserved
ELSEIF	SQL3	Keyword	PL/SQL	Keyword
END-EXEC	SQL92	Keyword		Keyword
END	SQL89	Keyword	PL/SQL	Keyword
EQUALS	SQL3	Keyword		Keyword
ESCAPE	SQL89	Keyword	Keyword	Keyword
EXCEPT	SQL92	Keyword	Keyword	Keyword
EXCEPTION	SQL92	Keyword	PL/SQL	Keyword
EXCEPTIONS			Keyword	
EXCLUSIVE			Reserved	Reserved
EXEC	SQL89	Keyword	Keyword	Keyword
EXECUTE	SQL92	Keyword	Keyword	Keyword
EXISTS	SQL89	Keyword	Reserved	Reserved
EXTERNAL	SQL92	Keyword		Keyword
EXTERNALLY			Keyword	
EXTRACT	SQL92	Partially Reserved		Keyword
FALSE	SQL92	Keyword	PL/SQL	Keyword
FETCH	SQL89	Keyword	PL/SQL	Keyword
FILE			Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
FIRST	SQL92	Keyword		Keyword
FLOAT	SQL89	Keyword	Reserved	Reserved
FOR	SQL89	Keyword	Reserved	Reserved
FOREIGN	SQL89	Keyword	Keyword	Keyword
FORTRAN	SQL89	Keyword	Keyword	Keyword
FOUND	SQL89	Keyword	Keyword	Keyword
FROM	SQL89	Partially Reserved	Reserved	Reserved
FULL	SQL92	Keyword		Keyword
FUNCTION	SQL3	Keyword	PL/SQL	Keyword
GENERAL	SQL3	Keyword		Keyword
GET	SQL92	Keyword		Keyword
GLOBAL	SQL92	Keyword		Keyword
GO	SQL89	Keyword	Keyword	Keyword
GOTO	SQL89	Keyword		Keyword
GRANT	SQL89	Keyword	Reserved	Reserved
GROUP	SQL89	Keyword	Reserved	Reserved
HAVING	SQL89	Keyword	Reserved	Reserved
HOUR	SQL92	Keyword		Keyword
IDENTIFIED			Reserved	Reserved
IDENTITY	SQL92	Keyword		Keyword
IF	SQL3	Keyword	PL/SQL	Keyword
IGNORE	SQL3	Keyword		Keyword
IMMEDIATE	SQL92	Keyword	Reserved	Reserved
IN	SQL89	Keyword	Reserved	Reserved
INCREMENT			Reserved	Reserved
INDEX			Reserved	Reserved
INDICATOR	SQL89	Partially Reserved	Keyword	Reserved
INITIAL			Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
INITIALLY	SQL92	Keyword		Keyword
INNER	SQL92	Keyword		Keyword
INOUT	SQL3	Keyword		Keyword
INPUT	SQL92	Keyword		Keyword
INSENSITIVE	SQL92	Keyword		Keyword
INSERT	SQL89	Keyword	Reserved	Reserved
INSTEAD	SQL3	Keyword		Keyword
INT	SQL89	Keyword	Keyword	Keyword
INTEGER	SQL89	Keyword	Reserved	Reserved
INTERSECT	SQL92	Keyword	Reserved	Reserved
INTERVAL	SQL92	Partially Reserved		Keyword
INTO	SQL89	Keyword	Reserved	Reserved
IS	SQL89	Keyword	Reserved	Reserved
ISOLATION	SQL92	Keyword		Keyword
JOIN	SQL92	Keyword		Keyword
KEY	SQL89	Keyword	Keyword	Keyword
LABEL				Reserved
LANGUAGE	SQL89	Keyword	Keyword	Keyword
LAST	SQL92	Keyword		Keyword
LEADING	SQL92	Keyword		Keyword
LEAVE	SQL3	Keyword		Keyword
LEFT	SQL92	Keyword		Keyword
LESS	SQL3	Keyword		Keyword
LEVEL	SQL92	Keyword	Reserved	Reserved
LIKE	SQL89	Keyword	Reserved	Reserved
LIMIT	SQL3	Keyword		Keyword
LOCAL	SQL92	Keyword		Keyword
LOCK			Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
LONG			Reserved	Reserved
LOOP	SQL3	Keyword	PL/SQL	Keyword
LOWER	SQL92	Partially Reserved		Keyword
MATCH	SQL92	Keyword		Keyword
MAX	SQL89	Partially Reserved	PL/SQL	Keyword
MAXEXTENTS			Reserved	Reserved
MIN	SQL89	Partially Reserved	PL/SQL	Keyword
MINUS			Reserved	Reserved
MINUTE	SQL92	Keyword		Keyword
MLSLABEL				Reserved
MLS_LABEL_FORMAT				Reserved
MODE			Reserved	Reserved
MODIFY	SQL3	Keyword	Reserved	Reserved
MODULE	SQL89	Partially Reserved	Keyword	Reserved
MONTH	SQL92	Keyword		Keyword
MOVE	SQL3	Keyword		Keyword
MULTISET	SQL3	Keyword		Keyword
NAMES	SQL92	Keyword		Keyword
NATIONAL	SQL92	Keyword		Keyword
NATURAL	SQL92	Keyword		Keyword
NCHAR	SQL92	Keyword		Keyword
NEW_TABLE	SQL3	Keyword		Keyword
NEXT	SQL92	Keyword	Keyword	Keyword
NO	SQL92	Keyword		Keyword
NOAUDIT			Reserved	Reserved
NOCOMPRESS			Reserved	Reserved
NONE	SQL3	Keyword	Keyword	Keyword
NOT	SQL89	Partially Reserved	Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
NOWAIT			Reserved	Reserved
NULL	SQL89	Partially Reserved	Reserved	Reserved
NULLIF	SQL92	Partially Reserved		Keyword
NUMBER			Reserved	Reserved
NUMERIC	SQL89	Keyword	Keyword	Keyword
NVL		Partially Reserved		
OCTET_LENGTH	SQL92	Partially Reserved		Keyword
OF	SQL89	Keyword	Reserved	Reserved
OFF	SQL3	Keyword	Keyword	Keyword
OFFLINE			Reserved	Reserved
OID	SQL3	Keyword		Keyword
OLD	SQL3	Keyword	Keyword	Keyword
OLD_TABLE	SQL3	Keyword		Keyword
ON	SQL89	Keyword	Reserved	Reserved
ONLINE			Reserved	Reserved
ONLY	SQL92	Keyword	Keyword	Keyword
OPEN	SQL89	Keyword	PL/SQL	Keyword
OPERATION	SQL3	Keyword		Keyword
OPERATORS	SQL3	Keyword		Keyword
OPTION	SQL89	Keyword	Reserved	Reserved
OR	SQL89	Keyword	Reserved	Reserved
ORDER	SQL89	Keyword	Reserved	Reserved
OTHERS	SQL3	Keyword	PL/SQL	Keyword
OUT	SQL3	Keyword	PL/SQL	Keyword
OUTER	SQL92	Keyword		Keyword
OUTPUT	SQL92	Keyword		Keyword
OVERLAPS	SQL92	Keyword		Keyword
PAD	SQL92	Keyword		Keyword

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
PARAMETER	SQL3	Keyword		Keyword
PARTIAL	SQL92	Keyword		Keyword
PASCAL	SQL89	Keyword	Keyword	Keyword
PCTFREE			Reserved	Reserved
PENDANT	SQL3	Keyword		Keyword
PLI	SQL89	Keyword		Keyword
POSITION	SQL92	Partially Reserved		Keyword
PRECISION	SQL89	Keyword	Keyword	Keyword
PREORDER	SQL3	Keyword		Keyword
PREPARE	SQL92	Keyword		Keyword
PRESERVE	SQL92	Keyword		Keyword
PRIMARY	SQL89	Keyword	Keyword	Keyword
PRIOR	SQL92	Keyword	Reserved	Reserved
PRIVATE	SQL3	Keyword	PL/SQL	Keyword
PRIVILEGES	SQL89	Keyword	Reserved	Reserved
PROCEDURE	SQL89	Keyword	PL/SQL	Keyword
PROTECTED	SQL3	Keyword		Keyword
PUBLIC	SQL89	Keyword	Reserved	Reserved
RAW			Reserved	Reserved
READ	SQL92	Keyword	Keyword	Keyword
READUP				Reserved
REAL	SQL89	Keyword	PL/SQL	Keyword
RECURSIVE	SQL3	Keyword		Keyword
REF	SQL3	Keyword	PL/SQL	Keyword
REFERENCES	SQL89	Keyword	Keyword	Keyword
REFERENCING	SQL3	Keyword	Keyword	Keyword
RELATIVE	SQL92	Keyword		Keyword
RENAME			Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
REPRESENTATION	SQL3	Keyword		Keyword
RESIGNAL	SQL3	Keyword		Keyword
RESOURCE			Reserved	Reserved
RESTRICT	SQL92	Keyword		Keyword
RETURN	SQL3	Keyword	PL/SQL	Keyword
RETURNS	SQL3	Keyword		Keyword
REVOKE	SQL92	Keyword	Reserved	Reserved
RIGHT	SQL92	Keyword		Keyword
ROLE	SQL3	Keyword	Keyword	Keyword
ROLLBACK	SQL89	Keyword	PL/SQL	Keyword
ROUTINE	SQL3	Keyword		Keyword
ROW	SQL3	Keyword	Reserved	Reserved
ROWID		Partially Reserved	Reserved	Reserved
ROWNUM		Partially Reserved	Reserved	Reserved
ROWS			Reserved	Reserved
SAVEPOINT	SQL3	Keyword	PL/SQL	Keyword
SCHEMA	SQL89	Keyword	PL/SQL	Keyword
SCROLL	SQL92	Keyword		Keyword
SEARCH	SQL3	Keyword		Keyword
SECOND	SQL92	Keyword		Keyword
SECTION	SQL89	Keyword	Keyword	Keyword
SELECT	SQL89	Keyword	Reserved	Reserved
SENSITIVE	SQL3	Keyword		Keyword
SEQUENCE	SQL3	Keyword	Keyword	Keyword
SESSION	SQL92	Keyword	Reserved	Reserved
SESSION_USER	SQL92	Partially Reserved	PL/SQL	Reserved
SET	SQL89	Keyword	Reserved	Reserved
SHARE			Reserved	Reserved

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
SIGNAL	SQL3	Keyword		Keyword
SIMILAR	SQL3	Keyword		Keyword
SIZE	SQL92	Keyword	Reserved	Reserved
SMALLINT	SQL89	Keyword	Reserved	Reserved
SOME	SQL89	Keyword	Keyword	Keyword
SPACE	SQL92	Keyword	PL/SQL	Keyword
SPECIFIC	SQL3	Keyword		Keyword
SQL	SQL89	Keyword	PL/SQL	Keyword
SQLCODE	SQL89	Keyword	PL/SQL	Keyword
SQLERROR	SQL89	Keyword	Keyword	Keyword
SQLEXCEPTION	SQL3	Keyword		Keyword
SQLSTATE	SQL92	Keyword		Keyword
SQLWARNING	SQL3	Keyword		Keyword
START			Reserved	Reserved
STRUCTURE	SQL3	Keyword		Keyword
SUBSTRING	SQL92	Partially Reserved		Keyword
SUCCESSFUL			Reserved	Reserved
SUM	SQL89	Partially Reserved	Keyword	Keyword
SYNONYM			Reserved	Reserved
SYSDATE		Partially Reserved	Reserved	Reserved
SYSTEM_USER	SQL92	Partially Reserved		Reserved
TABLE	SQL89	Keyword	Reserved	Reserved
TEMPLATE	SQL3	Keyword		Keyword
TEMPORARY	SQL92	Keyword	Keyword	Keyword
TEST	SQL3	Keyword		Keyword
THAN	SQL3	Keyword		Keyword
THEN	SQL92	Keyword	Reserved	Reserved
THERE	SQL3	Keyword		Keyword



**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
TIME	SQL92	Partially Reserved	Keyword	Keyword
TIMESTAMP	SQL92	Partially Reserved		Keyword
TIMEZONE_HOUR	SQL92	Keyword		Keyword
TIMEZONE_MINUTE	SQL92	Keyword		Keyword
TO	SQL89	Keyword	Reserved	Reserved
TRAILING	SQL92	Keyword		Keyword
TRANSACTION	SQL92	Keyword	Keyword	Keyword
TRANSLATE	SQL92	Partially Reserved		Keyword
TRANSLATION	SQL92	Keyword		Keyword
TRIGGER	SQL3	Keyword	Reserved	Reserved
TRIM	SQL92	Partially Reserved		Keyword
TRUE	SQL92	Keyword	PL/SQL	Keyword
TUPLE	SQL3	Keyword		Keyword
TYPE	SQL3	Keyword	PL/SQL	Keyword
UID			Reserved	Reserved
UNDER	SQL3	Keyword		Keyword
UNION	SQL89	Keyword	Reserved	Reserved
UNIQUE	SQL89	Keyword	Reserved	Reserved
UNKNOWN	SQL92	Keyword		Keyword
UPDATE	SQL89	Keyword	Reserved	Reserved
UPPER	SQL92	Partially Reserved		Keyword
USAGE	SQL92	Keyword		Keyword
USER	SQL89	Partially Reserved	Reserved	Reserved
USING	SQL92	Keyword	Keyword	Keyword
VALIDATE			Reserved	Reserved
VALUE	SQL92	Partially Reserved	Keyword	Reserved
VALUES	SQL89	Keyword	Reserved	Reserved
VARCHAR	SQL92	Keyword	Reserved	Reserved

---

**Table A-1 Comparison of Reserved Words and Semantics (Cont.)**

<b>Reserved Word</b>	<b>ANSI/ISO SQL Standard</b>	<b>Oracle Rdb Oracle Level1 Dialect</b>	<b>Oracle7</b>	<b>Oracle Rdb Oracle Level2<sup>1</sup> Dialect</b>
VARCHAR2			Reserved	Reserved
VARIABLE	SQL3	Keyword		Keyword
VARIANT	SQL3	Keyword		Keyword
VIRTUAL	SQL3	Keyword		Keyword
VARYING	SQL92	Keyword		Keyword
VIEW	SQL89	Keyword	Reserved	Reserved
VISIBLE	SQL3	Keyword		Keyword
WAIT	SQL3	Keyword		Keyword
WHEN	SQL92	Keyword	PL/SQL	Keyword
WHENEVER	SQL89	Keyword	Reserved	Reserved
WHERE	SQL89	Keyword	Reserved	Reserved
WHILE	SQL3	Keyword		Keyword
WITH	SQL89	Keyword	Reserved	Reserved
WITHOUT	SQL3	Keyword		Keyword
WORK	SQL89	Keyword	PL/SQL	Keyword
WRITE	SQL92	Keyword	PL/SQL	Keyword
WRITEDOWN				Reserved
WRITEUP				Reserved
YEAR	SQL92	Keyword		Keyword
ZONE	SQL92	Keyword		Keyword

<sup>1</sup> To be supported in a future version of Oracle Rdb.

---

---

# Index

## Symbols

---

(+) outer join, 3-4

## A

---

alias names, 3-2  
ALL predicate, 4-12  
ALTER FUNCTION statement, 6-8  
ALTER PROCEDURE statement, 6-8  
ALTER TABLE statement, 6-4  
ALTER TRIGGER statement, 6-8  
ALTER VIEW statement, 6-8  
ANY predicate, 4-12  
arithmetic operator, 2-1

## B

---

bit string data types, 1-15

## C

---

CASCADE keyword, 6-8  
CASE expression, 4-12  
CAST operator  
    concatenation, 2-2  
    conversions, 2-3  
CHAR data type, 1-11  
character data types, 1-11  
CHECK, 1-4  
CHECK constraints, 1-4  
check constraints  
    nondeterministic, 1-4  
    referencing, 1-4  
column names  
    qualified, 3-3  
columns  
    computed, 1-3  
    renaming, 4-15  
comment indicators, 3-3  
COMMENT ON statement, 6-7  
COMMIT statement, 6-1  
compiled objects  
    CREATE clause, 6-6  
    REPLACE clause, 6-6  
computed columns, 1-3  
concatenation  
    CAST operator, 2-2  
constraints, 1-3, 6-4  
    check, 1-3  
        nondeterministic, 1-4  
        referencing, 1-4  
    deferrable, 1-4, 1-5, 6-6  
    deferred, 1-4, 1-5  
    disabled  
        DISABLE clause, 6-5  
    enabled  
        ENABLE clause, 6-5  
    exceptions  
        EXCEPTIONS INTO clause, 6-5  
foreign key  
    cascade, 1-3  
    immediately deferred, 1-4  
    not deferrable, 1-5  
    not null, 1-3  
    primary key, 1-3  
referential delete  
    ON DELETE CASCADE clause, 6-4

- unique, 1-3, 1-5
- unique indexes
  - PRIMARY KEY keyword, 6-5
  - UNIQUE keyword, 6-5
- CONTAINING predicate, 4-13
- conversions
  - arithmetic expressions, 2-2
  - CAST operator, 2-3
  - dates, 2-3
- CREATE AS SELECT statement, 1-3
- CREATE DATABASE LINK statement, 1-8
- CREATE DOMAIN statement, 6-1
- CREATE FORCE statement, 1-7
- CREATE SEQUENCE statement, 1-5
- CREATE TABLE statement, 6-1, 6-3
- CREATE VIEW statement, 6-7
- created users, 1-9
- CURRVAL keyword, 4-3

## D

---

- data types, 1-11
  - bit string
    - RAW, 1-15
  - character
    - CHAR, 1-11
    - VARCHAR, 1-11
    - VARCHAR2, 1-11
  - date-time
    - DATE, 1-14
  - defaults, 6-2
  - long
    - LONG, 1-15
    - LONG RAW, 1-15
  - miscellaneous
    - MLSLABEL, 1-17
    - ROWID, 1-17
  - numeric
    - FLOAT, 1-12
    - NUMBER, 1-12
  - synonyms, 6-2
  - usage, 6-1
- database
  - attachments, 5-1
  - links, 1-8, 5-1

- multischema, 1-2
- single-schema, 1-2
- database link, 1-8
  - CREATE DATABASE LINK statement, 1-8
- database link names, 3-1
- database names, 3-1
- DATE data type, 1-14
- date-time data types, 1-14
- DBKEY keyword, 4-1
- Declare Variable statement, 6-1
- DECODE function, 4-11
- DEFERRABLE constraints, 1-4
- DEFERRED constraints, 1-4
- DELETE statement, 5-2, 5-5
- derived tables, 4-15
- dialects
  - Oracle Level1, A-1
  - Oracle Level2, A-1
  - Oracle Rdb, A-1
  - Oracle7, A-1
- distributed data, 1-9
  - schedule object, 1-10
  - Snapshot database object, 1-10
  - Snapshot Log database object, 1-10
  - transfer object, 1-10
- domains, 1-5
- DROP statement, 6-8
  - CASCADE keyword, 6-8
  - RESTRICT keyword, 6-8

## E

---

- EXCEPT operator, 4-14
- EXCEPTIONS INTO clause, 6-5
- EXISTS predicate, 4-12

## F

---

- FLOAT data type, 1-12
- FOREIGN KEY constraints
  - cascade, 1-4
- functions, 4-4 to 4-11

## G

---

global temporary tables, 1-3  
GROUP BY clause, 4-18

## I

---

IMMEDIATELY DEFERRED constraints, 1-4  
IN predicate, 4-12  
INSERT statement, 5-2  
    values and subqueries, 5-3  
INTERSECT operator, 4-14  
IS NULL predicate, 4-12

## J

---

JOIN operator  
    FULL, 2-2  
    INNER, 2-2  
    LEFT, 2-2  
    RIGHT, 2-2

## K

---

keywords, 3-5  
    SET KEYWORD RULES statement, 3-5  
    *See also* reserved words

## L

---

LEVEL keyword, 4-3  
LIKE predicate, 4-12  
LIMIT TO keyword, 4-14  
local temporary tables, 1-3  
LONG data type, 1-15  
long data types, 1-15  
LONG RAW data type, 1-15

## M

---

MINUS operator, 2-1, 4-14  
MLSLABEL data type, 1-17  
Multinational Character Sets, 1-17  
multischema databases, 1-2

## N

---

namespace, 1-10  
    of tables, 1-10  
    of views, 1-10  
NEXTVAL keyword, 4-3  
nonschema objects, 1-9  
    profiles, 1-9  
    roles, 1-9  
    SET QUERY statement, 1-9  
    users, 1-9  
NOT DEFERRABLE constraints, 1-5  
not equals operator, 3-4  
NOT NULL constraints, 1-4  
NULL constraints, 1-4  
NUMBER data type, 1-12  
numeric data types, 1-12

## O

---

object names, 3-1  
object recompilation  
    ALTER FUNCTION statement, 6-8  
    ALTER PROCEDURE statement, 6-8  
    ALTER TRIGGER statement, 6-8  
    ALTER VIEW statement, 6-8  
ON DELETE CASCADE clause, 6-4  
operators, 2-1  
    arithmetic, 2-1  
    CAST, 2-2  
    concatenation, 2-2  
    EXCEPT, 2-1  
    INTERSECT, 2-1  
    JOIN  
        FULL, 2-2  
        INNER, 2-2  
        LEFT, 2-2  
        RIGHT, 2-2  
    MINUS, 2-1, 4-14  
    not equals, 3-4  
    set, 2-1  
    UNION, 2-1  
    UNION ALL, 2-1  
OPTIMIZE keyword, 4-14  
Oracle dialect, A-1

- Oracle Level1 dialect, A-1
- Oracle Level2 dialect, A-1
- Oracle Rdb dialect
  - Oracle Level1, A-1
  - Oracle Level2, A-1
- ORDER BY clause, 4-17
- outer join (+), 3-4

## P

---

- populating tables, 1-3
- predicates, 4-12
  - ALL, 4-12
  - ANY, 4-12
  - CONTAINING, 4-13
  - EXISTS, 4-12
  - IN, 4-12
  - IS NULL, 4-12
  - LIKE, 4-12
  - SINGLE, 4-13
  - SOME, 4-12
  - STARTING WITH, 4-13
- preface
  - Send Us Your Comments, vii
- PRIMARY KEY constraints, 1-4
- profiles, 1-9
- pseudocolumns, 4-1
  - recursive unions
    - LEVEL keyword, 4-3
  - row addresses
    - DBKEY keyword, 4-1
    - ROWID keyword, 4-1
  - rows returned
    - ROWNUM keyword, 4-2
  - sequences
    - CURRVAL keyword, 4-3
    - NEXTVAL keyword, 4-3

## Q

---

- qualified names
  - column names, 3-3
  - table names, 3-2
- quoted names, 3-1

## R

---

- RAW data type, 1-15
- recursive unions
  - LEVEL keyword, 4-3
- RENAME statement, 6-9
- renaming database objects
  - columns, 4-15
    - RENAME statement, 6-9
  - sequences, 6-9
  - synonyms, 6-9
  - tables, 6-9
  - views, 6-9
- reserved words, 3-5, A-1 to A-14
  - built-in functions, 3-6
  - qualified, 3-5
  - quoted, 3-5
- RESTRICT keyword, 6-8
- roles, 1-9
- row addresses
  - DBKEY keyword, 4-1
  - ROWID keyword, 4-1
- ROWID data type, 1-17
- ROWID keyword, 4-1
- ROWNUM keyword, 4-2, 4-14
- rows returned
  - ROWNUM keyword, 4-2

## S

---

- schedule object, 1-10
- schemas, 1-1
  - Distributed Option for Rdb, 1-3
  - multischema databases, 1-2
  - single-schema databases, 1-2
- scope of table names, 4-16
- select expression, 4-14
  - derived tables, 4-15
  - GROUP BY clause, 4-18
  - keywords
    - LIMIT TO, 4-14
    - OPTIMIZE, 4-14
    - ROWNUM, 4-14
  - ORDER BY clause, 4-17
  - recursive unions, 4-14, 4-15

- renaming columns, 4-15
- scope of table names, 4-16
- SELECT list
  - DISTINCT keyword, 4-15
  - UNIQUE keyword, 4-15
- set operators
  - EXCEPT, 4-14
  - INTERSECT, 4-14
  - MINUS, 4-14
  - UNION, 4-14
  - UNION ALL, 4-14
- SELECT list, 4-15
- Send Us Your Comments
  - boilerplate, vii
- sequences, 1-5
  - CREATE SEQUENCE statement, 1-5
  - CURRVAL keyword, 4-3
  - NEXTVAL keyword, 4-3
  - renaming, 6-9
- SET KEYWORD RULES statement, 3-5
- set operators, 2-1
- SET QUERY statement, 1-9
- SINGLE predicate, 4-13
- single-schema databases, 1-2
- Snapshot database object, 1-10
- Snapshot Log database object, 1-10
- snapshots
  - COMMENT ON statement, 6-7
- SOME predicate, 4-12
- SQL names, 3-1
  - alias, 3-1, 3-2
  - database, 3-1
  - database link, 3-1
  - objects, 3-1
  - quoted, 3-1
- STARTING WITH predicate, 4-13
- subqueries
  - DELETE statement, 5-2
  - INSERT statement, 5-2
  - UPDATE statement, 5-2
- synonyms, 1-6
  - renaming, 6-9
  - See also* views

## T

---

- table names
  - scope of, 4-16
- tables, 1-3
  - altering, 6-4
  - COMMENT ON statement, 6-7
  - CREATE AS SELECT statement, 1-3
  - creating, 6-3
  - derived, 4-15
  - names
    - qualified, 3-2
  - namespace, 1-10
  - populating, 1-3
  - renaming, 6-9
  - temporary
    - global, 1-3
    - local, 1-3
  - with computed columns, 1-3
- temporary tables
  - global, 1-3
  - local, 1-3
- transfer object, 1-10
- transfers
  - specific to Distributed Option for Rdb, 1-2
- triggers, 1-6

## U

---

- UNION ALL operator, 4-14
- UNION operator, 4-14
- UNIQUE constraints, 1-4, 1-5
- UPDATE statement, 5-2
  - bulk updates
    - column with UNIQUE constraint, 5-4
  - correlated subqueries, 5-5
  - multiple columns, 5-2
  - SET clause, 5-2
  - single column, 5-2
  - values and subqueries, 5-3

## V

---

- VARCHAR data type, 1-11
- VARCHAR2 data type, 1-11
- views, 1-7

COMMENT ON statement, 6-7  
CREATE FORCE statement, 1-7  
CREATE VIEW statement, 6-7  
namespace, 1-10  
renaming, 6-9  
*See also* DBKEY keyword