10

Creating and Managing Tables

ORACLE®

Objectives

At the end of this lesson, you will be able to:

- Describe the main database objects
- Create tables
- Describe the datatypes that can be used when specifying column definition
- Alter table definitions
- Drop, rename, and truncate tables

Database Objects

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Generates primary key values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Naming Conventions

- Must begin with a letter
- Can be 1–30 characters long
- Must contain only A–Z, a–z, 0–9, _, $, and #
- Must not duplicate the name of another object owned by the same user
- Must not be an Oracle8 Server reserved word

The CREATE TABLE Statement

- You must have :
  - CREATE TABLE privilege
  - A storage area

```
CREATE TABLE [schema.]table
             (column datatype [DEFAULT expr];
```

- You specify:
  - Table name
  - Column name, column datatype, and column size

Referencing Another User’s Tables

- Tables belonging to other users are not in the user’s schema.
- You should use the owner’s name as a prefix to the table.

The DEFAULT Option

- Specify a default value for a column during an insert.

```
… hiredate DATE DEFAULT SYSDATE, …
```

- Legal values are literal value, expression, or SQL function.
- Illegal values are another column’s name or pseudocolumn.
- The default datatype must match the column datatype.

Creating Tables

- Create the table.

```
SQL> CREATE TABLE dept
  2           (deptno  NUMBER(2),
  3            dname   VARCHAR2(14),
  4            loc     VARCHAR2(13));
Table created.
```

- Confirm table creation.

```
 SQL> DESCRIBE dept
```

```
Name                           Null?    Type
------------------------------ -------- ---------
DEPTNO                         NOT NULL NUMBER(2)
DNAME                                   VARCHAR2(14)
LOC                                     VARCHAR2(13)
```

ORACLE®

Querying the Data Dictionary

- Describe tables owned by the user.

```
SQL> SELECT   *
  2  FROM     user_tables;
```

- View distinct object types owned by the user.

```
SQL> SELECT   DISTINCT object_type
  2  FROM     user_objects;
```

- View tables, views, synonyms, and sequences owned by the user.

```
SQL> SELECT   *
  2  FROM     user_catalog;
```

Datatypes

Datatype	Description
VARCHAR2(*size*)	Variable-length character data
CHAR(*size*)	Fixed-length character data
NUMBER(*p*,*s)*	Variable-length numeric data
DATE	Date and time values
LONG	Variable-length character data up to 2 gigabytes
CLOB	Single-byte character data up to 4 gigabytes
RAW and LONG RAW	Raw binary data
BLOB	Binary data up to 4 gigabytes
BFILE	Binary data stored in an external file; up to 4 gigabytes

Creating a Table by Using a Subquery

- Create a table and insert rows by combining the CREATE TABLE statement and AS *subquery* option.

```
CREATE TABLE table
        [column(, column...)]
AS subquery;
```

- Match the number of specified columns to the number of subquery columns.
- Define columns with column names and default values.

Creating a Table by Using a Subquery

```
SQL> CREATE TABLE   dept30
  2  AS
  3      SELECT    empno, ename, sal*12 ANNSAL, hiredate
  4      FROM      emp
  5      WHERE     deptno = 30;
Table created.
```

```
SQL> DESCRIBE dept30
```

```
Name                            Null?    Type
------------------------------- -------- -----
EMPNO                           NOT NULL NUMBER(4)
ENAME                                    VARCHAR2(10)
ANNSAL                                   NUMBER
HIREDATE                                 DATE
```

The ALTER TABLE Statement

Use the ALTER TABLE statement to:

- Add a new column
- Modify an existing column
- Define a default value for the new column

```
ALTER TABLE table
ADD         (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
MODIFY      (column datatype [DEFAULT expr]
            [, column datatype]...);
```

Adding a Column

DEPT30

EMPNO	ENAME	ANNSAL	HIREDATE
7698	BLAKE	34200	01-MAY-81
7654	MARTIN	15000	28-SEP-81
7499	ALLEN	19200	20-FEB-81
7844	TURNER	18000	08-SEP-81
. . .			

New column

JOB

"…add a new column into DEPT30 table…"

DEPT30

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
. . .				

ORACLE®

Adding a Column

- You use the ADD clause to add columns.

```
SQL> ALTER TABLE dept30
  2  ADD         (job VARCHAR2(9));
Table altered.
```

- The new column becomes the last column.

```
    EMPNO ENAME         ANNSAL HIREDATE   JOB
--------- ---------- --------- --------- ----
     7698 BLAKE          34200 01-MAY-81
     7654 MARTIN         15000 28-SEP-81
     7499 ALLEN          19200 20-FEB-81
     7844 TURNER         18000 08-SEP-81
...
6 rows selected.
```

Modifying a Column

- You can change a column’s datatype, size, and default value.

```
ALTER TABLE   dept30
MODIFY         (ename VARCHAR2(15));
Table altered.
```

- A change to the default value affects only subsequent insertions to the table.

Dropping a Table

- All data in the table is deleted.
- Any pending transactions are committed.
- All indexes are dropped.
- You *cannot* roll back this statement.

```
SQL> DROP TABLE dept30;
Table dropped.
```

Changing the Name of an Object

- To change the name of a table, view, sequence, or synonym, you execute the RENAME statement.

```
SQL> RENAME dept TO department;
Table renamed.
```

- You must be the owner of the object.

Truncating a Table

- The TRUNCATE TABLE statement:
  - Removes all rows from a table
  - Releases the storage space used by that table

```
SQL> TRUNCATE TABLE department;
Table truncated.
```

- Cannot roll back row removal when using TRUNCATE
- Alternatively, remove rows by using the DELETE statement

Adding Comments to a Table

- You can add comments to a table or column by using the COMMENT statement.

```
SQL> COMMENT ON TABLE emp
  2  IS 'Employee Information';
Comment created.
```

- Comments can be viewed through the data dictionary views.
  - ALL_COL_COMMENTS
  - USER_COL_COMMENTS
  - ALL_TAB_COMMENTS
  - USER_TAB_COMMENTS

Summary

Statement	Description
CREATE TABLE	Creates a table
ALTER TABLE	Modifies table structures
DROP TABLE	Removes the rows and table structure
RENAME	Changes the name of a table, view, sequence, or synonym
TRUNCATE	Removes all rows from a table and releases the storage space
COMMENT	Adds comments to a table or view

Practice Overview

- Creating new tables
- Creating a new table by using the CREATE TABLE AS syntax
- Modifying column definitions
- Verifying that the tables exist
- Adding comments to a tables
- Dropping tables
- Altering tables