

## CREATING AND MANAGING TABLES

### -- CHAPTER 10 --

- Database Objects: An Oracle database can contain multiple data structures. Each structure should be outlined in the database design so that it can be created during the build stage of database development.
  - Table: Stores data
  - View: Subset of data from one or more tables
  - Sequence: Generates primary key values
  - Index: Improves the performance of some queries
  - Synonym: Gives alternative names of objects
- Oracle8 Table Structures
  - Tables can be created at any time, even while users are using the database
  - You do not need to specify the size of any table. The size is ultimately defined by the amount of space allocated to the database as a whole. It is important, however, to estimate how much space a table will use over time.
  - Table structure can be modified online.
- Naming Conventions:
  - Must begin with a letter and must be no longer than 30 characters.
  - 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 Oracle Server reserved word.
    - Note that names are case insensitive.
- The CREATE TABLE statement: You must have CREATE TABLE privilege and Storage Area...
  - Must specify Table Name
    - Must specify 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

```
SQL> r
1 CREATE TABLE dept
2 (deptno NUMBER(2),
3  dname VARCHAR2(14),
4*  loc VARCHAR2(13))
```

Table created.

```
SQL> describe dept
Name                               Null?      Type
```

```

-----
DEPTNO                                NUMBER (2)
DNAME                                  VARCHAR2 (14)
LOC                                     VARCHAR2 (13)

```

- Tables in the Oracle Database
  - USER TABLES
    - Collection of tables created and maintained by the user
    - Contain user information
  - DATA DICTIONARY
    - Collection of tables created and maintained by the Oracle Server
    - Contain database information
- All data dictionary tables are owned by the SYS user. The base tables are rarely accessed by the user because the information in them is not easy to understand. Therefore, users typically access data dictionary views because the information is presented in a format that is easier to understand. Information stored in the data dictionary include names of the Oracle Server users, privileges granted to users, database object names, table constraints, and auditing information.
- There are four categories of data dictionary views, each category has a distinct previews which reflects their intended use.
  - USER\_ - These views contain information about objects owned by the user
  - ALL\_ - These views contain information about all of the tables (object tables and relational tables) accessible to the user.
  - DBA\_ - These views are restricted views. These views can be accessed only by people who have been assigned the role DBA.
  - V\$\_ - These views contain information about dynamic performance views, database server performance and locking.
- Querying the Data Dictionary
  - SELECT \* FROM user\_tables;

```

SQL> select table_name
      2  from user_tables;

```

```

TABLE_NAME
-----
CONTACT
CUSTOMER
DEPT
ZIP_TABLE

```

- SELECT DISTINCT object\_type FROM user\_objects;

```

SQL> select distinct object_type
      2  from user_objects;

```

```

OBJECT_TYPE
-----
TABLE

```

- SELECT \* FROM user\_catalog;

```
SQL> select *
      2  from user_catalog;
```

TABLE_NAME	TABLE_TYPE
CONTACT	TABLE
CUSTOMER	TABLE
DEPT	TABLE
ZIP_TABLE	TABLE

- Datatypes

- VARCHAR2(size)
  - Variable-length character data (max size must be specified).
- CHAR(size)
  - Fixed-length character data of length size bytes.
- NUMBER(p,s)
  - Number having precision p and scale s (The precision is the total number of decimal digits, and the scale is the number of digits to the right of the decimal point. The precision can range from 1 to 38 and the scale can range from -84 to 127.
- DATE
  - Date and time values between January 1, 4712 B.C. and December 31, 9999 A.D.
- 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 of length size (A maximum size must be specified. Maximum size is 2000).
- 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. You must make sure to match the number of specified columns with the number of subquery columns. You must define columns with columns names and default values.

```
SQL> CREATE TABLE dept30 AS
      2  SELECT tempo, ename, sal*12 ANNSAL, hiredate
      3  from emp
      4  WHERE deptno = 30;
```

Table created.

```
SQL> describe dept30
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)

ANNSAL	NUMBER
HIREDATE	DATE

```
SQL> select * from dept30;
```

EMPNO	ENAME	ANNSAL	HIREDATE
7499	ALLEN	12000	20-FEB-81
7521	WARD	12000	22-FEB-81
7698	BLAKE	24000	01-MAY-81
7844	TURNER	12000	08-SEP-81

- The previous table created a table containing details of all the employees working in department 30. Notice that all the data in dept30 came from table emp.
- The ALTER TABLE statement is used to: Add a new column, Modify an existing column, or Define a default value for the new column.
- You use the ADD clause to add columns to an existing table:

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

Table altered.

```
SQL> select * from dept30;
```

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7499	ALLEN	12000	20-FEB-81	
7521	WARD	12000	22-FEB-81	
7698	BLAKE	24000	01-MAY-81	
7844	TURNER	12000	08-SEP-81	

- You can add or modify columns, but you cannot drop them from a table. You cannot specify where the column is to appear. The new columns becomes the last column.
- Modifying a Column: You can change the column's data type, size, and value.

```
SQL> describe dept30
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
ANNSAL		NUMBER
HIREDATE		DATE
JOB		VARCHAR2(9)

```
SQL> r
```

```
1 alter table dept30
2* modify (job VARCHAR2(20))
```

Table altered.

```
SQL> describe dept30
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
ANNSAL		NUMBER
HIREDATE		DATE
JOB		VARCHAR2(20)

EMPNO	NOT NULL	NUMBER (4)
ENAME		VARCHAR2 (10)
ANNSAL		NUMBER
HIREDATE		DATE
JOB		VARCHAR2 (20)

- **Dropping a Column:** You use the **DROP COLUMN** clause to drop columns you no longer need from the table.

```
SQL> alter table dept30
2 drop column job;
```

Table altered.

```
SQL> describe dept30;
```

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

- **SET UNUSED Option:** You use the **SET UNUSED** option to mark one or more columns as unused. You use the **DROP UNUSED COLUMNS** option to remove the columns that are marked as **UNUSED**.

```
SQL> alter table dept30
2 set unused column hiredate;
```

Table altered.

```
SQL> select * from dept30;
```

EMPNO	ENAME	ANNSAL
-----	-----	-----
7499	ALLEN	12000
7521	WARD	12000
7698	BLAKE	24000
7844	TURNER	12000

- In this book, it is not shown how to revert to undoing this table modification. Note that column hiredate was not dropped, merely marked as unused.
- **Dropping a Table:** All data and structure in the table is deleted. Any pending transactions are committed. All indexes are dropped. You cannot rollback this statement.
- **NOTE:** Oracle will not question your request to drop a table and the modification is permanent and irreversible! It seems weird that you are not asked to confirm your request before making such a drastic modification, but such is the case!!!
- **Changing the Name of an Object:** To change the name of a table, view, sequence, or synonym, you execute the **RENAME** statement.

```
SQL> rename dept30 to Departamento;
```

Table renamed.

```
SQL> describe Departamento;
Name                               Null?    Type
-----
ENAME                               VARCHA2 (10)
DEPTNO                              NUMBER (2)
ANNSAL                              NUMBER
MGR                                  NUMBER (4)
```

- The TRUNCATE TABLE statement: Removes all rows from a table, Releases the storage space used by that table. You cannot rollback row removal when using TRUNCATE. Of course, you can remove rows by using the DELETE statement.

```
SQL> TRUNCATE table Departamento;
```

Table truncated.

```
SQL> select * from Departamento;
```

**no rows selected**

- Finally, comments can be added to a table or column by using the COMMENT statement. Such comments can be viewed through the data dictionary views.

```
SQL> COMMENT ON TABLE Departamento
2 IS 'This table is completely empty -
3 since it was TRUNCATED';
```

Comment created.