LAB 6: Creating and Managing Tables

Objectives

After completing this lesson, you should be able to do the following:

- ✓ Create tables, Describe the data types
- ✓ Alter table definitions, Drop, rename, and truncate tables
- ✓ Add new rows to a table, Updating Rows in a Table, and Remove existing rows from a table

Lesson Aim

- ➢ You will learn how to create, alter, drop tables; add, update, and remove rows of a table

Naming Rules

Name database tables and columns according to the standard rules for naming any Oracle database object:

• Table names and column names must begin with a letter and be 1–30 characters long.

• Names must contain only the characters A–Z, a–z, 0–9, _ (underscore), $, and # (legal characters, but their use is discouraged).

1. CREATE TABLE dept80
   (deptno NUMBER(2),
   dname VARCHAR2(14),
   loc VARCHAR2(13));
2. DESCRIBE dept

Data Types

Data type Description

CHAR Data Type

The CHAR data type stores strings that have a fixed length. This means that the value takes up the same number of bytes, regardless of the value it holds.

When you create a table with a column defined as CHAR, you need to specify a length. The length must be between 1 and 2000. This length is the number of characters to store in the field.

The main thing to know about CHAR values is that if you insert or update a value that is less than this length, then the value is padded with blank characters up to the length of the column.

For example, let's say you have a column with a data type of CHAR(10). You insert the value of "Tree", which is only 4 characters long.

The field will contain the value of "Tree" but it will be padded with six space characters on the right hand side, turning it into a value of "Tree ".

If you wanted to insert a value of "Large Oak Tree", Oracle would return an error. This is because the value inserted is too large for the column (14 character value into a 10 character column).

When to use CHAR: There should be no reason to use the CHAR data type, as it is similar to a VARCHAR2 and it's better to be consistent.

VARCHAR2 Data Type

The VARCHAR2 data type stores alphanumeric values in variable-length strings. This means, unlike the CHAR data type, the length is variable.

When you create a column with the VARCHAR2 data type, you specify a maximum length between 1 and 4000 bytes. Only the characters in the value you insert or update are stored.

For example, if a column is defined as VARCHAR2(10), then it can store up to 10 characters. If I insert a value of "Tree", which is 4 characters, then the value stored is "Tree". No spaces are added, and the value does not need to be 10 characters.

Using a VARCHAR2 column will often save space when compared to CHAR columns.

Why is it called VARCHAR2 and not VARCHAR?

The VARCHAR is a standard data type, and isn't used in Oracle SQL. When Oracle wanted to implement their own version of VARCHAR, rather than changing the implementation of VARCHAR, they created VARCHAR2.

So, if you come from another database system such as SQL Server or MySQL, don't use VARCHAR – use VARCHAR2.

When to use VARCHAR2: If you're storing text values that can vary in length (which is probably most of the time).

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)

For example, defining a number column like this will round all values to the nearest thousand:
NUMBER (12, 3);
So, with all of these different ways you can specify a NUMBER field, how would a value look with each of these possibilities?
Let's take a look. What if we tried to insert the value 12,345,678.9012 into a NUMBER column?

Number Column Specified	**Value Stored As**
NUMBER	12,345,678.9012
NUMBER(10,2)	12,345,678.90
NUMBER(*,1)	12,345,678.9
NUMBER(10)	12,345,678
NUMBER(5)	Error – exceeds precision
NUMBER(10,1)	12,345,678.9
NUMBER(10,-3)	12,346,000

So, you can see that the same number value can be stored in many different ways, depending on how you define your column.

DATE Date and time values to the nearest second between January 1, 4712 B.C., and A.D. December 31, 9999. The default input and output format of a DATE value is DD-MON-YY. For example, for 12th October 2016: "12-OCT-16"

More Character Data Types

Data Type	Description	MAX Size
CHAR(size)	Fixed length character with a length of size.	2000 bytes. Default and minimum is 1 byte
NCHAR(size)	Fixed length national character with a length of size.	2000 bytes. Default and minimum is 1 byte
VARCHAR	Deprecated and only used for backward compatibity.	
VARCHAR2(size)	Variable length character string with a maximum length of size bytes	4000 bytes. Minimum is 1 byte
NVARCHAR2(size)	Variable length national character string with a maximum length of size bytes	4000 bytes. Minimum is 1 byte
LONG	Variable length character string. Larger than VARCHAR2. Deprecated	2 GB
RAW(size)	Raw binary data of length size	2000 bytes
LONG RAW	Raw binary data of variable length. Deprecated	2 GB

The ALTER TABLE Statement

After you create a table, you may need to change the table structure because you omitted a column or your column definition needs to be changed or you need to remove columns. You can do this by using the

ALTER TABLE statement.

3. ALTER TABLE dept80
   ADD (job_id VARCHAR2(9));

4. ALTER TABLE dept80
   MODIFY (last_name VARCHAR2(30));

5. ALTER TABLE dept80
   DROP COLUMN job_id;

6. Dropping a Table
   DROP TABLE dept80;

Changing the Name of an Object

7. RENAME dept80 TO detail_dept;

Add new rows to a table

```
INSERT INTO departments(department_id, department_name,
manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
```

Updating Rows in a Table

```
UPDATE employees
SET department_id = 70
WHERE employee_id = 113;
```

Remove existing rows from a table

```
DELETE FROM departments
WHERE department_name = 'Finance';
```

Truncating a Table

Another DDL statement is the `TRUNCATE TABLE` statement, which is used to remove all rows from a table and to release the storage space used by that table. When using the `TRUNCATE TABLE` statement, you cannot rollback row removal.

Syntax

```
TRUNCATE TABLE table;
```

```
TRUNCATE TABLE detail_dept;
Table truncated.
```

See the names of tables owned by the user.

```
SELECT table_name
FROM user_tables;
```

Creating a Table by Using a Subquery

```
CREATE TABLE dept80
```

```
AS
SELECT employee_id, last_name,
salary*12 ANNSAL,
hire_date
FROM employees
WHERE department_id = 80;
```

Practice

1. Create the `EMPLOYEE` table based on the following table instance chart.

Name	Null?	Type
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

2. Insert the following data

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750

3. Modify the `EMP` table to allow for longer employee last names. Confirm your modification.

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

4. Confirm that `EMPLOYEE` tables are stored in the data dictionary.
5. Change the last name of employee 3 to *Drexler*.
6. Change the salary to 1000 for all employees with a salary less than 900.
7. Delete *Betty Dancs* from the `MY_EMPLOYEE` table.
8. Rename the `EMPLOYEES` table as `EMP80`
9. Drop the `FIRST_NAME` column from the `EMPLOYEE` table. Confirm your modification by checking the description of the table.
10. Create the `EMPLOYEE2` table based on the structure of the `EMPLOYEE` table. Include only the `EMPLOYEE_ID`, `FIRST_NAME`, `LAST_NAME`, `SALARY`.