

## 3.GETTING STARTED WITH ORACLE8i

- Creating a table
- Datatypes
- Displaying table definition using DESCRIBE
- Inserting rows into a table
- Selecting rows from a table
- Editing SQL buffer
- Summary
- Exercises

### Creating a Table

A Table is a collection of rows and columns. Data in relational model is stored in tables. Let us create a table first. Then we will understand how to store data into table and retrieve data from the table.

Before a table is created the following factors of a table are to be finalized.

- What data table is supposed to store.
- The name of the table. It should depict the content of the table.
- What are the columns that table should contains
- The name, data type and maximum length of each column of the table.
- What are the rules to be implemented to main data integrity of the table.

The following is an example of creation of COURSES table. We actually have six tables in the application that we use throughout the book. We will discuss more about all the tables in the next chapter. But for now, let us create COURSES table and understand how to use basic SQL commands.

The following CREATE TABLE command is used to create COURSES table.

```
SQL> create table COURSES
  2  ( ccode      varchar2(5),
  3    name       varchar2(30),
  4    duration   number(3),
  5    fee        number(5),
  6    prerequisite varchar2(100)
  7  );
```

Table Created

The above command creates a table called COURSES. This table contains 5 columns. We will discuss about rules to be implemented in this table in the next chapter, where we will recreate this table with all the required rules.

For the time being I want to keep things simple. That is the reason why I am not taking you into constraint and remaining.

Well, we have created our first table. If command is successful, Oracle responds by displaying the message *Table Created*.

### Rules to be followed for names

The following are the rules to be followed while naming an Oracle Object. These rules are applicable for name of the table and column.

- ❑ The name must begin with a letter - **A-Z** or **a-z**.
- ❑ Letters, digits and special characters – underscore (**\_**), **\$** and **#** are allowed.
- ❑ Maximum length of the name is **30** characters.
- ❑ It must not be an SQL reserved word.
- ❑ There should not be any other object with the same name in your account.

---

**Note:** A table can contain up to 1000 columns in Oracle8 or above, whereas in Oracle7 a table can contain only 254 columns.

---

### Datatypes

Each column of the table contains the datatype and maximum length, if it is length is applicable. Datatype of the column specifies what type of data can be stored in the column.

The datatype VARCHAR2 is to store strings that may have different number of characters, NUMBER is used to store numbers. The maximum length, which is given in parentheses after the datatype, specifies how many characters (or digits) the column can store at the most. For example, column VARCHAR2 (20) would mean it can store up to 20 characters.

Table-1 lists out datatypes available in Oracle8i along with what type of data can be stored and maximum length allowed.

Datatype	Description
<b>VARCHAR2(len)</b>	Can store up to <b>len</b> number of characters. Each character would occupy one byte. Maximum width is 4000 characters.
<b>VARCHAR(len)</b>	Same as VARCHAR2. But use VARCHAR2 as Oracle might change the usage of VARCHAR in future releases.
<b>CHAR(len)</b>	Fixed length character data. If <b>len</b> is given then it can store up to <b>len</b> number of characters. Default width is 1. String is padded on the right with spaces until string is of <b>len</b> size. Maximum width is 2000.

<b>NUMBER</b>	Can store numbers up to 40 digits plus decimal point and sign.
<b>NUMBER (p ,s)</b>	<b>P</b> represents the maximum significant digits allowed. <b>S</b> is the number of digits on the right of the decimal point.
<b>DATE</b>	Can store dates in the range 1-1-4712 B.C to 31-12-4712 AD.
<b>LONG</b>	Variable length character values up to 2 gigabytes. Only one LONG column is allowed per table. You cannot use LONG datatype in functions, WHERE clause of SELECT, in indexing and subqueries.
<b>RAW and LONG RAW</b>	Equivalent to VARCHAR2 and LONG respectively, but used for storing byte-oriented or binary data such as digital sound or graphics images.
<b>CLOB, BLOB, NCLOB</b>	Used to store large character and binary objects. Each can accommodate up to 4 gigabytes. We will discuss more about it later in this book.
<b>BFILE</b>	Stores a pointer to an external file. The content of the file resides in the file system of the operation system. Only the name of the file is stored in the column.
<b>ROWID</b>	Stores a unique number that is used by Oracle to uniquely identify each row of the table.
<b>NCHAR (size)</b>	Same as CHAR, but supports national language.
<b>NVARCHAR2 (size)</b>	Same as VARCHAR2, but supports national language.

**Table 1:** Oracle Datatypes.

## Displaying table definition using DESCRIBE

You can display the structure of a table using SQL\*PLUS command DESCRIBE. It displays then name, datatype and whether the column can store null value for each column of the table.

The following is the syntax of DESCRIBE command.

```
DESC[RIBE] objectname
```

Displays the column definitions for the specified object. The object may be a table, view, synonym, function or procedure.

To display the structure of COURSES table, enter:

```
SQL> DESC COURSES
```

Name	Null?	Type
CCODE	NOT NULL	VARCHAR2 (5)
NAME		VARCHAR2 (30)
DURATION		NUMBER (3)
FEE		NUMBER (5)
PREREQUISITE		VARCHAR2 (100)

End of ebook preview

Download the full PDF tutorial from the link below :

[Click Here](#)