Creating Tables

A Table is a database object that is used to store data. There are several other objects available that we will be discussing as a later date.

A table can be created at any time, even while the user is using the database. We do not need to specify the overall size of the table since, in most cases, it will be constantly growing. We can also ‘modify’ the table structure on-line.

Naming Convention

These naming convention apply to the table name and also the column names.

1) Must begin with a letter
2) Can be up to 30 characters in length.
3) Must contain only A – Z, a – z, 0 – 9, _ , $, and #.
4) Must not duplicate the name of another object owned by the same user.
5) Cannot be a reserved word.

Naming Guideline

1) Use descriptive names for table and column names.
2) Name the same entity consistently in different table. Example: the department number column is called DEPTNO in both the EMP table and the DEPT table.
3) Names ARE case Sensitive meaning that EMP is different then eMP or eMp.

Table creation requirement

This statement is classed as a DDL (Data Definition Language) statement.

The first thing we must specify is the Table Name.

We then define the columns with:

Column Name

Column Datatype

Column Size

Schema

A schema is a collection of objects. Schema objects are the logical structure that directly refer to the data in a database.

Schema objects include tables, views, synonyms, sequences, stored procedures, indexes, clusters and database links.

If a table does not belong to your schema you have to prefix it with the owners name much like you do in MySQL and MSSQL when you refer to the tables mane.

Data Types

VARCHAR2(size)	Variable-length Character data.
CHAR(size)	Fixed-length Character data.
NUMBER(p,s)	Variable-length numeric data.
DATE	Date and time values Format DD-MON-YY
LONG	Variable-length Character data up to 2 gigabytes.
CLOB	Single-byte Character data up to 4 gigabytes
RAW & LONG RAW	Raw binary data
BLOB	Binary data up to 4 gigabytes

Syntax for Creating Tables

We are going to create a table called SILLYPUDDY and this is a mythical table we will use in our examples.

```
CREATE TABLE sillypuddy
    (deptno      NUMBER(2),
     dname       VARCHAR2(14),
     loc         VARCHAR2(13));
```

The reserved words are in UPPER CASE and the user defined words are in lower case. (Just for readability)

Altering Tables

We have created a table called SILLYPUDDY that we want to add a column to in our schema. When we do this we have to provide the column name and also the data type.

```
ALTER TABLE gpage.sillypuddy
ADD column  location VARCHAR2(20);
```

We have added the column location and it will have the value placed in it of NULL.

Altering a Column

If for some reason we need to alter or modify a column in the table we do that with either ALTER or MODIFY, depending on the dialect we are using.

MODIFY is used on ORACLE and MySQL.

ALTER is used in MSSQL.

```
ALTER TABLE sillypuddy
MODIFY location VARCHAR(10);
```

```
ALTER TABLE sillypuddy
ALTER COUMN location VARCHAR(10);
```

Renaming a Tables

We have created a table called SILLYPUDDY that we want to rename the table to something more meaningful.

```
RENAME gpage.sillypuddy to gpage.moremeaningful;
```

We have changed to name of the table and all of the data now belong to the object of moremeaningful.

MSSQL:

```
EXEC SP_RENAME ‘sillypuddy’, ‘moremeaningful’;
```

Remember to include your schema name on all commands.

Removing Tables

We have created a table called SILLYPUDDY and changed its name to moremeaningful and we want to remove from our schema. Remember - this is a mythical table we will use in our examples.

```
DROP TABLE gpage.moremeaningful;
```

The table structure and also the content are removed. Use this command with care. Be certain to include your name on all commands.

Removing Tables content

There may be times when we want to remove the data from the table BUT want to keep the structure of a table. Remembering that if we drop the table the content and structure are removed. However if we to remove the data only but maintain the structure we can issue this command.

```
DELETE *
FROM moremeaningful;
```

The data is removed but the structure remains as it is presently defined.

Next Lecture

In the next lecture we will cover the DML statements to Insert, Update and Delete information from tables.