

TO MODIFY COLUMNS

Several conditions apply to modifying the datatypes of existing columns or to add columns to a table in the database. The general thumb rule is that increases are generally OK, while decreases are usually a little trickier. Some examples of increases that are generally acceptable are listed as follows:

- Increases to the size of a VARCHAR2 or CHAR column.
- Increases in size of a NUMBER column.
- Increasing the number of columns in the table.

The following list details the allowable operations that decrease various aspects of the database:

- Reducing the number of columns in a table (empty table only)
- Reducing the size of a NUMBER column (empty column for all rows only)
- Reducing the length of a VARCHAR2 or CHAR column (empty column for all rows only)
- Changing the datatype of a column (empty column for all rows only)

The Modify option changes the following of the existing column

- Datatype
- Column width
- Constraints i.e. DEFAULT, NOT NULL or NULL.

There are certain restrictions while making these modifications:

- The type and/or size of a column can be changed, if every row for the column is NULL.
- An existing column can be modified to NOT NULL only if it has a non-null value in every row.

The syntax of the MODIFY option is

Alter table <tablename > [Modify < columnname >.....];

Examples

- Modify the job column of the emp table by increasing its width to varchar2 (40).

```
SQL> ALTER TABLE emp  
MODIFY job varchar2 (40);
```

- Modify the job column of the emp table by increasing its width to varchar2 (35) and applying the constraint of NOT NULL.

```
SQL> ALTER TABLE emp  
MODIFY job varchar2 (35) NOT NULL;
```

The DROP option

This option removes the constraints from a table. When a constraint is dropped, any associated index with that constraint (if there is one) is also dropped.

The syntax is

```
Alter table < tablename >  
[DROP <constraints> .....] ;
```

Example

- Remove the primary key constraint from the emp table.

```
SQL> ALTER TABLE emp  
DROP primary key;
```

- Remove the constraint on the deptno field of the dept table.

```
SQL> ALTER TABLE dept DROP constraint pk_deptno ;
```

Drop column option

Alter table command can also be used to drop the existing columns of the tables, but this option is available only in Oracle 8i or versions after that.

Syntax

```
ALTER TABLE <table_name> DROP COLUMN <column_name>;
```

Example

- Drop the column ename of emp table.

```
SQL> ALTER TABLE emp DROP COLUMN ename;
```

Removing Tables

In order to remove a table from the database, the drop table command must be executed.

The syntax is

```
DROP table <tablename >;
```

Example

```
SQL> DROP TABLE emp;
```

However, dropping tables may not always be that easy. Recall from the earlier lesson that when you disable constraints like primary keys that have foreign-key constraints in other tables depending on their existence, you may have some errors. The same thing happens when you try to drop the table that has a primary key referenced by enabled foreign keys in another table. If you try to drop a table that has other tables' foreign keys referring to it, the following error will ensue:

ORA-02266: unique/primary keys in table referenced by enabled foreign keys

When there are foreign-key constraints on other tables that reference the table to be dropped, then you can use on delete cascade option. The constraints in other tables that refer to the table being dropped are also dropped with cascade option. There are usually some associated objects that exist in a database along with the table. These objects may include the index that is created by the primary key or the unique constraint that is associated with columns in the table. If the table is dropped, Oracle automatically drops any index associated with the table as well.

```
SQL> DROP TABLE dept  
CASCADE CONSTRAINTS;
```

Alternately, you can disable or drop the foreign key in the other table first and then issue the drop table statement without the cascade constraints option. However, with this method you run the risk that many other tables having foreign keys that relate back to the primary key in the table you want to drop will each error out, one at a time, until you disable or drop every foreign-key constraint referring to the table. If there are several, your drop table activity may get extremely frustrating.

Available tables as Data Dictionary

User tables are tables created by the user such as student. There is another collection of tables, which are owned by the SYS user in the Oracle database known as data dictionary. These tables are maintained and created by the Oracle Server and contains information about the database. Information stored in the data dictionary include name of the Oracle Server users, privileges granted to users, database objects names, table constraints and other information. You can query the following data dictionary tables to view various database objects owned by you .The data dictionary tables frequently used are:

- USER_TABLES
 - USER_OBJECTS
 - USER_CATALOG
-

Flash Back

The table data structure stores the data and it is composed of rows and columns. A table can also represent a relation between two entities. First step before creating tables is to design them so that it gives a meaningful picture. One must follow rules to create a table as the name of the should not be a SQL reserve word , not more than 30 characters,

must contain only the characters A-Z, a-z, 0-9, _, \$, # . the name of the table is case insensitive.

The table can be created with the create statement and the table can be created from the existing table also which may or may not contain data. We can impose number of constraints on the table as a whole as well as on the columns also. these constraints provides data integrity and are divided into two categories:

- Column constraints(applied on one column)
- Table constraints(applied on more than one column)

The different types of constraints are;

- Not null constraint
- Unique constraint
- Primary key constraint
- Default constraint
- Foreign key constraint
- Check constraint

The user can see the names of the constraints in the user_constraints table. Finally, if we want to change the structure of the table we have the ALTER table command. With the alter statement we can add column or constraints, modify the datatypes or to add more columns, and drop the column or table itself.
