Database Design and Implementation of a Secure Online Shopping System

Brandon Russell, Joshua Oluwadamilare, Roha Seyoum, Tinuola Okusanya

University of Maryland University College

Group 3 Project Part 2: DBST670

Professor: Selwyn Igwe

April 21, 2019

Table of Contents

ALTER SQL Command

Add Column

Lengthen Column

Shorten Column

Change Data Type from NUMBER to CHAR

Change Data Type from CHAR to NUMBER

Validation

CREATE SQL Command

Create New Table

Populate Data into the Newly Created Table

Validation

Reestablish Referential Integrity

Drop Existing Table

Rename Table

Design for a Test of Restructuring Existing Objects

Step 1) Create Backup

Step 2) Create New Table

Step 3) Populate Data into Newly Created Table

Step 4) Alter Newly Created Table

Step 5) Drop Existing Table

Step 6) Rename Newly Created Table

Step 7) Confirm Removal of Existing Table

# ALTER SQL Command

Below are SQL commands that are applied to alter a table by adding a column, lengthening a column or shortening a column and changing the data type of an existing column from NUMBER to CHAR and then CHAR to NUMBER.

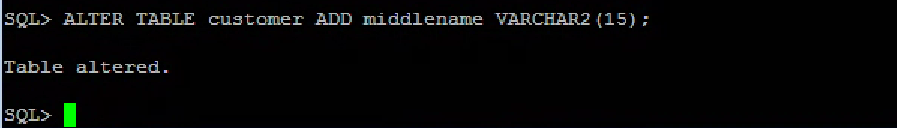
**Add Column**

SQL> connect dbauser/abc123;

Connected.

SQL> ALTER TABLE customer ADD middlename VARCHAR2(15);

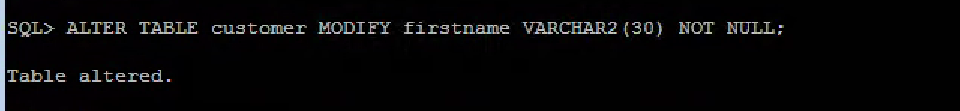
Table altered.



**Lengthen Column**

SQL> ALTER TABLE customer MODIFY firstname VARCHAR2(30) NOT NULL;

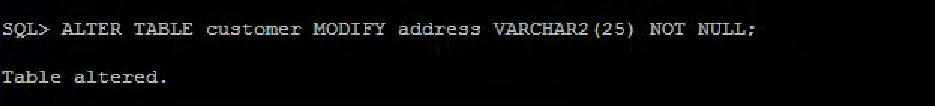
Table altered.



**Shorten Column**

SQL> ALTER TABLE customer MODIFY address VARCHAR2(25) NOT NULL;

Table altered.

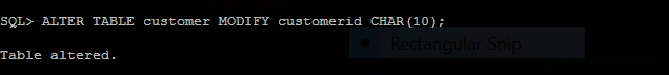


**Change Data Type from NUMBER to CHAR**

A data type for a customerid column changed from NUMBER TO CHAR as follows:

SQL> ALTER TABLE customerid MODIFY customertid CHAR (10);

Table altered.

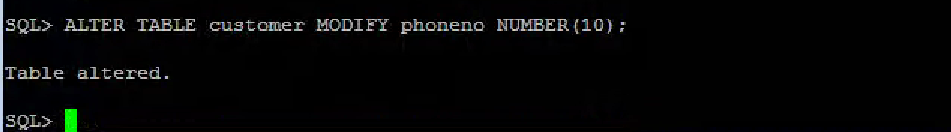


**Change Data Type from CHAR to NUMBER**

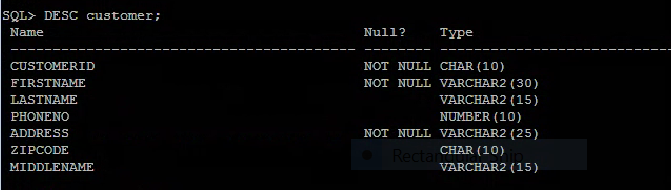
A data type for phoneno column changed from NUMBER TO CHAR as follows:

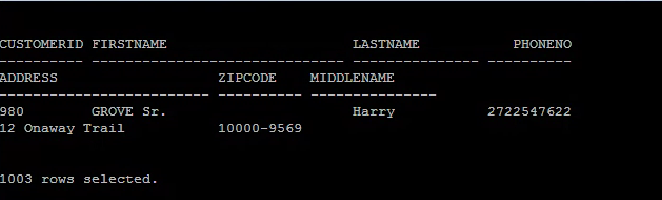
SQL> ALTER TABLE customer MODIFY phoneno NUMBER (10);

Table altered.



**Validation**

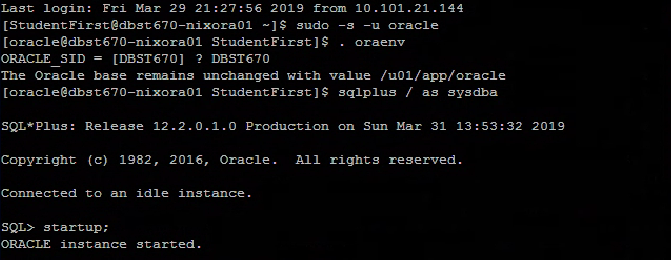




**CREATE SQL Command**

Below are SQL commands that are applied to create another table like an existing table without one column. Then, populate the data from the previously existing table to the newly created table. Thereafter, reestablished referential integrity with the primary key on the newly created table. Finally, dropped the previously existing table and renamed the newly created table.

Create New Table



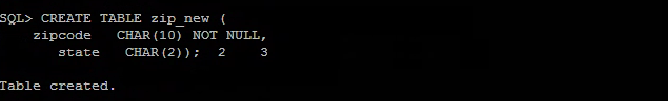
The new table zip\_new like an existing table without the column “city” is created.

SQL> CREATE TABLE zip\_new (

zipcode CHAR (10) NOT NULL,

state CHAR (2)); 2 3

Table created.

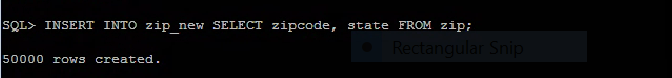


Populate Data into Newly Created Table

Loaded data into the newly created table from previously existing table “zip” as follows:

SQL> INSERT INTO zip\_new SELECT zipcode, state FROM zip;

50000 rows created.

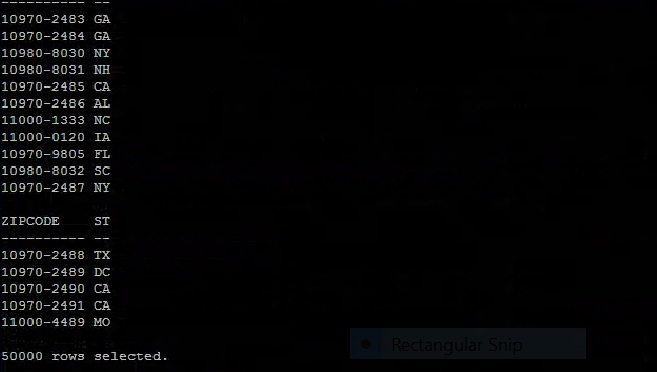


**Validation**

Verified the values are inserted into the new table

SQL> select \* from zip\_new;



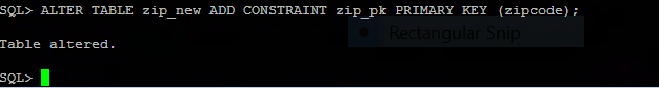


Reestablish Referential Integrity

Referential integrity with primary reestablished for the newly created table as follows:

SQL> ALTER TABLE zip\_new ADD CONSTRAINT zip\_pk PRIMARY KEY (zipcode);

Table altered.



Drop Existing Table

The previously existing table “zip” dropped as follows:

SQL> DROP TABLE zip CASCADE CONSTRAINTS;

Table dropped.



**Rename Table**

The newly created table “zip\_new” renamed as follows:

SQL> RENAME zip\_new TO zip;

Table renamed.



**Design for a Test of** **Restructuring Existing Objects**

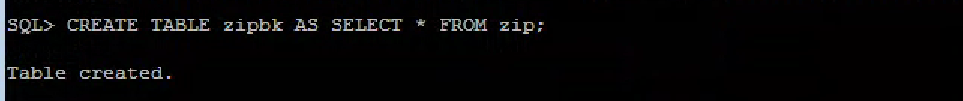
Below are important steps and SQL commands implemented to test the restructuring of existing objects in the above section.

**Step 1) Create Backup**

Backup the existing table (create back up) as follows:

SQL> CREATE TABLE zipbk AS SELECT \* FROM zip;

Table created.



Step 2) Create New Table

Create the new table zip\_new

SQL> CREATE TABLE zip\_new(

2 zipcode CHAR (10) NOT NULL,

3 state CHAR (2));

Table created.

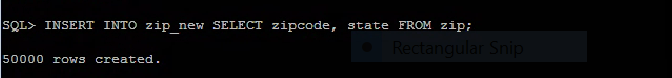


Step 3) Populate Data into the Newly Created Table from the Existing Table

Loaded data into the newly created table from previously existing table as follows:

SQL> INSERT INTO zip\_new SELECT zipcode, state FROM zip;

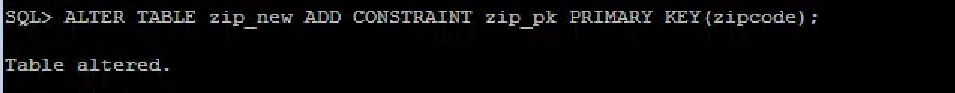
50000 rows created.



Step 4) Alter the Newly Created Table

SQL> ALTER TABLE zip\_new ADD CONSTRAINT zip\_pk PRIMARY KEY (zipcode);

Table altered.

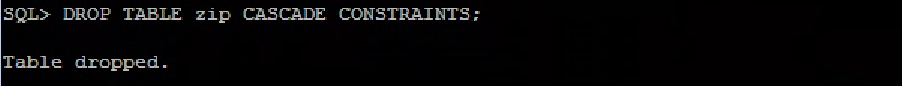


**Step 5) Drop Existing Table**

Previously existing table is dropped.

SQL> DROP TABLE zip CASCADE CONSTRAINTS;

Table dropped.



**Step 6) Rename Newly Created Table**

Renamed the newly created table to the original one as follows:

SQL> RENAME zip\_new TO zip;

Table renamed.



Step 7) Confirm the Removal of the Existing Table

SQL> DESC zip;

Name Null? Type

----------------------------------------- -------- ----------------------------

ZIPCODE NOT NULL CHAR (10)

STATE CHAR (2)

