

## Master–Detail with Zeos, Firebird and Lazarus or Delphi

This is a simple tutorial that allows to create a relationship Master-Detail between 2 tables of a database using the components Zeos (6.6.6) and Firebird (2.1.3).

It is made in Lazarus (0.9.31), using the distribution CodeTyphon (1.4) of Lazarus (which includes the Zeos components) and with Windows XP.

It should also work well with Delphi, with Lazarus in Linux and with other versions of Firebird, although it has not been tested.

In this manual all the steps will be detailed to carry out. It is important to go carrying out the process step for step.

The manual was made in Lazarus in Spanish, hence it is possible that some image or the name of some menu is not similar to that of Lazarus in another language (like English).

Create a new application.  
Menu File/New and select Application.

Save the application in a folder.  
Menu File/Save all...

To record a text file in the same folder in which stayed the application in the previous step. The text file name is 'create.sql.'  
You can do it using the program 'Notepad' or any other text editor.  
That file will include the SQL orders that will create the tables and the information for the database example that will be used in this manual.  
**The text of this file is at the end of this manual.**

Select the lash of Zeos component (Zeos Access) in the palette of components.



On the Form, add one component TZconnection ( **Zconnection1**) and one component TZSQLProcessor ( **ZSQLProcessor1** ).  
In the componente ZSQLProcessor1, in the property **Connection**, write/select Zconnection1.

On the Form, add one component Tbutton ( **Button1**) (lash Standard).  
In the property **Caption**, write '&Create DB / Crear BD' (without quotation marks).  
Double click on the component. When the code appears for the event Button1Click (Procedure **Tform1.Button1Click**), add this code:

```
ZConnection1.Connected := false ;  
  
    // Create file with Database Firebird, ...  
    // Crea archivo con la Base de Datos Firebird, ...  
if FileExists( aa )
```

```

    then DeleteFile( aa );
ZConnection1.Database := aa ;
ZConnection1.Protocol := 'firebird-2.1';
ZConnection1.Properties.Add (
  'CreateNewDatabase=CREATE DATABASE ' + QuotedStr ( aa )
+ ' USER ' + QuotedStr ('sysdba') + ' PASSWORD ' + QuotedStr ('masterkey')
+ ' PAGE_SIZE 4096'
// + ' DEFAULT CHARACTER SET ISO8859_1'
);

ZConnection1.User := 'sysdba' ;
ZConnection1.Password := 'masterkey' ;

ZConnection1.Connect;

    // ... create the tables and add the information
    // ... crea las tablas y añade la información.
ZSQLProcessor1.Script.LoadFromFile('create.sql');
ZSQLProcessor1.Execute;

ShowMessage( 'The DB has been created' + #13+ 'La BD ha sido creada');
//ZQuery1.Open;
//ZQuery2.Open;
//if sender = button1
// then DBGrid1.SetFocus;

```

On the Form, add one component Tbutton (**Button2**) (lash Standard).  
In the property **Caption**, write '&Exit / Salir' (without quotation marks).

Double click on the component. When the code appears for the event Button2Click (Procedure **Tform1.Button2Click**), add this code:

```
Close ;
```

Click on the Form (it would be usually Form1) and, in the object inspector, click on the Events lash.

Later, on the event **OnCreate**, do click in the button with three points '...' and add this code:

```

aa := ExtractFilePath ( Application.ExeName ) + 'cities.fdb' ;

//if FileExists( aa )
//then
// begin
//   ZConnection1.Database := aa ;
//   ZQuery1.Open;
//   ZQuery2.Open;
// end
//else
// Button1Click( nil );

```

In the section 'Var' (before 'implementation'), add this code to declare the public variable aa:

```
aa : string ;
```

Save the changes of the application (**Ctrl+s**) and compile (**F9**).  
When you see the application running, do click on the button 'Create DB / Crear BD'.  
If everything goes well, a message will appear indicating that BD has been created.  
Close the application making click in the button 'Exit / Salir'  
The file 'cities.fdb' has been created in the folder of the application.

In the component **ZConnection1**:

- In the property **Database**, do click in the button with 3 points '...' and select the database 'cities.fdb' created in the previous step.
- In the property **Password**, write 'masterkey' (without quotation marks).
- In the property **Protocol**, select 'Firebird -2.1.' It is made in this way because we are using Firebird 2.1.3, if another version of Firebird was used, it would be necessary to select the protocol that corresponds to the version of Firebird.
- In the property **User**, write 'sysdba' (without quotation marks).

Select the lash of Zeos component.

Add on the Form one component TZQuery (**ZQuery1**) and one component TZUpdateSQL (**ZUpdateSQL1**).

In the component Zquery1:

- in the property **Connection**, write /select Zconnection1.
- in the property **UpdateObject**, write /select ZUpdateSQL1.
- in the property **SQL**, do click in the button with 3 points '...' and write the following code: 'Select \* from COUNTRY' (without quotation marks).

Do click with the right button of the mouse on the component **Zquery1** and select the first option of the menu ('Edit fields...'):



An empty window will appear. Do click in the button with a sign '+.'

Then a window will appear with the names of all the fields of the table COUNTRY. Select all the fields and do click in the button 'Create'.

It close the window and the names of the fields will appear in the window that initially was empty:

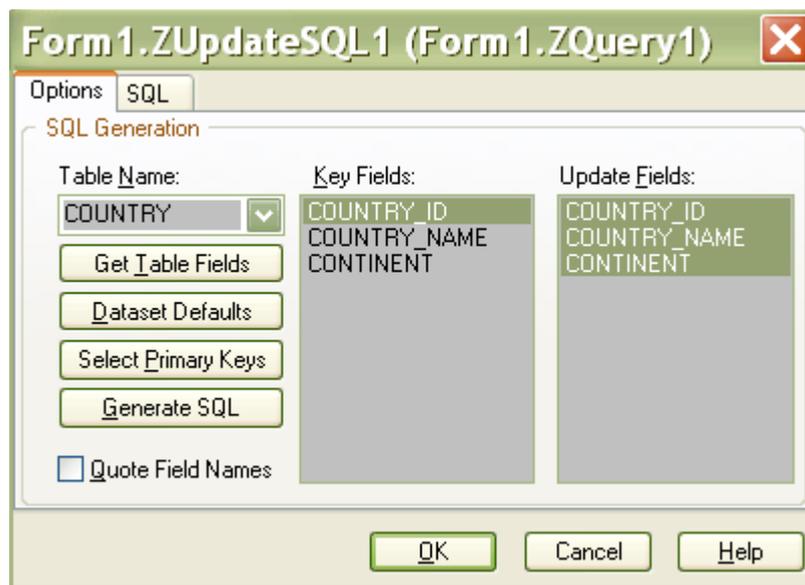


Close the window.

Do click with the right button of the mouse on the component **ZUpdateSQL1** y select the first option of the menu ('UpdateSql editor...'):



A window will appear. Do click on the field **COUNTRY\_ID** of 'Key fields:'. The window will be seen like in this image:



Later, do click on the button 'Generate SQL'.

The tab 'SQL' will be activated and will show the code automatically generated.

The code of 'Modify' and 'Delete' must be modified.

In the code of **Modify**, the final code will be:

```
UPDATE COUNTRY SET
  COUNTRY_NAME = :COUNTRY_NAME,
  CONTINENT = :CONTINENT
WHERE
  COUNTRY_ID = :OLD_COUNTRY_ID
```

In the code of **Delete**, the final code will be:

```
DELETE FROM COUNTRY
WHERE
  COUNTRY_ID = :OLD_COUNTRY_ID
```

When the modifications have been made, do click on the button 'OK'.

Add one component TDataSource (**Datasource1**) (lash Data Access).  
In the property **DataSet** write / select Zquery1.

Select the lash of Zeos component.  
Add on the Form one component TZQuery (**ZQuery2**) and one component TZUpdateSQL (**ZUpdateSQL2**).

In the component Zquery2:

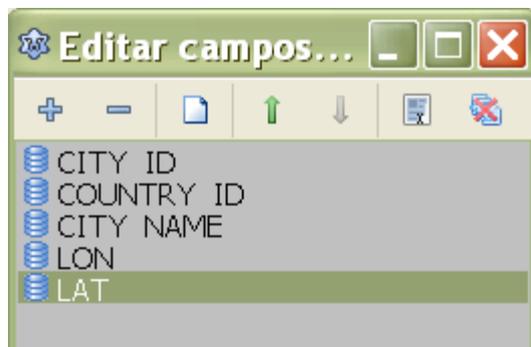
- in the property **Connection**, write /select Zconnection1.
- in the property **DataSource**, write /select Datasource1.
- in the property **UpdateObject**, write /select ZUpdateSQL2.
- in the property **SQL**, do click in the button with 3 points '...' and write the following code:  
'Select \* from CITY where COUNTRY\_ID=:COUNTRY\_ID' (without quotation marks).

Do click with the right button of the mouse on the component **Zquery2** and select the first option of the menu ('Edit fields...'):

An empty window will appear. Do click in the button with a sign '+'.

Then a window will appear with the names of all the fields of the table CITY. Select all the fields and do click in the button 'Create'.

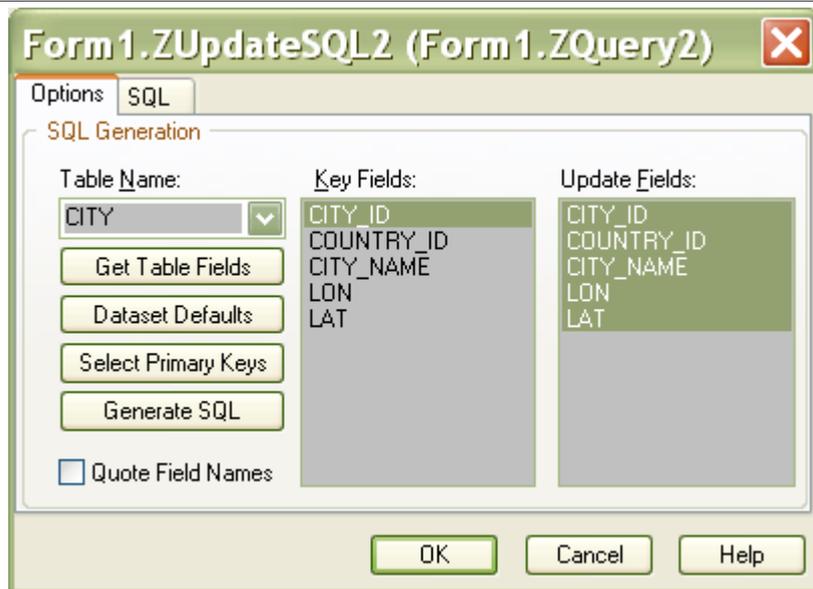
It close the window and the names of the fields will appear in the window that initially was empty:



Close the window.

Do click with the right button of the mouse on the component **ZUpdateSQL1** y select the first option of the menu ('UpdateSql editor...'):

A window will appear. Do click on the field CITY\_ID of 'Key fields:'. The window will be seen like in this image:



Later, do click on the button 'Generate SQL'.  
 The lash 'SQL' will be activated and will show the code automatically generated.  
 The code of 'Modify' and 'Delete' must be modified.

In the code of **Modify**, the final code will be:

```
UPDATE CITY SET
  CITY_ID = :CITY_ID,
  COUNTRY_ID = :COUNTRY_ID,
  CITY_NAME = :CITY_NAME,
  LON = :LON,
  LAT = :LAT
WHERE
  CITY_ID = :OLD_CITY_ID
```

In the code of **Delete**, the final code will be:

```
DELETE FROM CITY
WHERE
  CITY_ID = :OLD_CITY_ID
```

When the modifications have been made, do click on the button 'OK'.

Add one component TDataSource (**Datasource2**) (lash Data Access).  
 In the property **DataSet** write / select Zquery2.

On the Form, add one component TDBGrid (**DBGrid1**) and one component TDBNavigator (**DBNavigator1**) (lash Data Controls).  
 In the property **DataSource** of these two components, write / select Datasource1.

On the Form, add one component TDBGrid (**DBGrid2**) and one component TDBNavigator (**DBNavigator2**) (lash Data Controls).  
 In the property **DataSource** of these two components, write / select Datasource2.  
 In the component DBNavigator2, in the property **Visible**, write / select False.

The comments are code lines that begin with the characters '//.'

In the procedure **TForm1.FormCreate** (where previously a code was added with comments), to remove the characters '// in the comments of the last lines of the code.

You can make it by hand or you can select the text of that code and, in the menu Edit, do click in 'Uncomment selection' (the text in English may be different, because I use the Spanish translation of Lazarus).

Do click on the **Form** and, in the object inspector, do click in the Events lash.

Later, in the event **OnDestroy**, do click in the button with 3 points '...' and add this code:

```
ZConnection1.Connected:= false;
```

Select **DBGrid1** doing click on it and, maintaining the key 'Shift' pressed, do click on **DBGrid2**. Both components will be selected on this way.

Press **F11** and, in the Object Inspector, do click in the lash Events.

Then, in the event **OnEnter**, do click on the button with three points '...' and add this code:

```
if sender = DBGrid1
then
begin
  DBNavigator1.Visible := true ;
  DBNavigator2.Visible := false ;
end
else
begin
  DBNavigator2.Visible := true ;
  DBNavigator1.Visible := false ;
end;
```

Do click on **Zquery1** and, in the Object Inspector, do click in the lash Events.

Then, in the event **BeforeDelete**, do click on the button with three points '...' and add this code:

```
// Before erasing the registration of a country, it is necessary
// to erase the cities that belong to that country
// Antes de borrar el registro de un país, hay que borrar las ciudades
// que pertenecen a ese país
ZSQLProcessor1.Script.Clear;
ZSQLProcessor1.Script.Add(
  'Delete from CITY where COUNTRY_ID = '
  + IntToStr( ZQuery1COUNTRY_ID.AsInteger ) + ';' ) ;

ZSQLProcessor1.Execute ;
```

In the procedure **TForm1.Button1Click** (where previously a code was added with comments), to remove the characters '// in the comments of the last 4 lines of the code.

Save the changes (**Ctrl+s**) and compile the application (**F9 ó Ctrl+F9**).

In this application, when do click on the button 'Create DB / Crear BD', will be erased the previous DB 'cities.fdb' (if it existed) and a new DB is created with the initial same data. Hence, you can erase and modify data making tests and, when new data are needed, you can create again the initial DB.

In the table Detail (CITY), you can erase any register and to modify any field. If you modify the information of the field COUNTRY\_ID, the final result should be a value that exists in the table Teacher; otherwise, an error will appear.

En la tabla Maestro (COUNTRY), se puede modificar la información de cualquier campo, excepto la información del campo COUNTRY\_ID. Y se puede borrar cualquier registro. Cuando se borra un registro en la tabla Maestro (COUNTRY), anteriormente se borran todos los registros de la tabla Detalle (CITY), que dependen de él (por el campo COUNTRY\_ID); ese proceso se realiza en el procedimiento **Tform1.ZQuery1BeforeDelete** .

In the table Master (COUNTRY), you can modify the information of any field, except the information of the field COUNTRY\_ID. And you can erase any register. When you erase a register in the table Master (COUNTRY), previously the application erase all the registers of the table Detal (CITY) that depend of him (for the field COUNTRY\_ID); that process is carried out in the procedure **Tform1.ZQuery1BeforeDelete**.

This manual is made for beginners and/or newbies. The author is also a beginner and newbie. Hence, it is very possible that small, medium and/or big errors exist. Even this way, it is expected that it is useful.

For suggestions, corrections, improvements, etc., write to: **paratodo99@gmail.com**

In the address [www.paratodo99.000a.biz/pas](http://www.paratodo99.000a.biz/pas) you can download the complete code of this application.

Thank you for your interest.

This is the content of the file 'create.sql':

```
/*-----*/
CREATE TABLE COUNTRY
(
  COUNTRY_ID Integer NOT NULL,
  COUNTRY_NAME Varchar(18),
  CONTINENT Varchar(12),
  PRIMARY KEY (COUNTRY_ID)
);

CREATE TABLE CITY
(
  CITY_ID Integer NOT NULL,
  COUNTRY_ID Integer,
  CITY_NAME Varchar(15),
  LON Integer,
  LAT Integer,
  PRIMARY KEY (CITY_ID)
);
ALTER TABLE CITY
ADD FOREIGN KEY (COUNTRY_ID)
REFERENCES COUNTRY (COUNTRY_ID);

/*---- COUNTRIES -----*/
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 1,'Turkey' , 'Asia' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 2,'Russia' , 'Europe' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 3,'China' , 'Asia' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 4,'Kazajstan' , 'Asia' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 5,'Sweden' , 'Europe' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 6,'Spain' , 'Europe' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 7,'India' , 'Asia' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 8,'Australia' , 'Oceania' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES ( 9,'Indonesia' , 'Asia' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (10,'Ethiopia' , 'Africa' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (11,'Egypt' , 'Africa' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (12,'Nigeria' , 'Africa' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (13,'Uganda' , 'Africa' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (14,'New Zealand' , 'Oceania' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (15,'United States' , 'America' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (16,'Argentina' , 'America' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (17,'Fidji' , 'Oceania' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (18,'Paraguay' , 'America' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (19,'Ecuador' , 'America' );
INSERT INTO COUNTRY (COUNTRY_ID,COUNTRY_NAME, CONTINENT) VALUES (20,'Mexico' , 'America' );

/*----- CITIES -----*/
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 1,1, 'Ankara' , -32,39 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 2,1, 'Antalya' , -30,37 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 3,1, 'Denizli' , -29,37 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 4,1, 'Canakkale' , -26,40 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 5,2, 'Moscow' , -37,56 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 6,2, 'Kursk' , -36,52 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 7,2, 'Ekaterinburgo' , -60,57 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 8,2, 'Vologda' , -51,62 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 9,3, 'Beijing' , -117,40 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 10,3, 'Datong' , -114,40 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 11,3, 'Shanghai' , -122,32 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 12,3, 'Baotou' , -110,40 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 13,4, 'Guriev' , -52,47 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 14,4, 'Sonal' , -70,49 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 15,4, 'Aktiubinsk' , -57,50 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 16,4, 'Temirtau' , -72,50 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 17,5, 'Stockholm' , -18,58 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 18,5, 'Uppsala' , -17,59 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 19,5, 'Uddevalla' , -12,58 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 20,6, 'Madrid' , 3,41 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 21,6, 'Gijon' , 5,43 );
```

```

INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 22,6, 'Barcelona' , -2,41 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 23,6, 'Malaga' , 4,37 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 24,7, 'Kanpur' , -80,26 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 25,7, 'Calcuta' , -88,22 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 26,7, 'Bombay' , -72,18 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 27,7, 'Delhi' , -77,28 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 28,8, 'Sydney' , -151,-34 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 29,8, 'Perth' , -116,-32 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 30,8, 'Melbourne' , -145,-37 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 31,8, 'Brisbane' , -153,-27 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 32,9, 'Yakarta' , -106,-6 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 33,9, 'Sarmi' , -138,-2 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 34,9, 'Surabaya' , -112,-7 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 35,9, 'Tobo' , -130,-3 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (36,10, 'AddisAbeba' , -39,9 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (37,10, 'Mitsiwa' , -39,15 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (38,10, 'Gorgora' , -37,12 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (39,10, 'Mekele' , -39,13 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (40,11, 'Cairo' , -31,30 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (41,11, 'Assuan' , -33,24 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (42,11, 'Luxor' , -32,25 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (43,11, 'Alexandria' , -29,31 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (44,12, 'Lakes' , -4,6 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (45,12, 'Abuja' , -7,9 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (46,12, 'Sokoto' , -5,13 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (47,12, 'Maiduguri' , -13,12 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (48,13, 'Kampala' , -32,0 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (49,13, 'Hoima' , -31,1 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (50,13, 'Gulu' , -32,2 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (51,13, 'Moroto' , -34,2 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (52,14, 'Hamilton' , -175,-37 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (53,14, 'Dunedin' , -170,-46 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (54,14, 'Christchurch' , -172,-43 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (55,14, 'Wellington' , -174,-41 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (56,15, 'Filadelfia' , 75,40 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (57,15, 'Columbus' , 83,40 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (58,15, 'Seattle' , 122,48 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (59,15, 'LosAngeles' , 118,34 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (60,16, 'BuenosAires' , 58,-34 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (61,16, 'Cordoba' , 64,-31 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (62,16, 'Castelli' , 60,-26 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (63,16, 'Rosario' , 60,-33 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (64,16, 'Buena Esperanza' , 65,-34 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (65,17, 'Suva' , -178,-17 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (66,17, 'Nadi' , -177,-17 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (67,17, 'Sigatoka' , -177,-17 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (68,18, 'Asunción' , 57,-25 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (69,18, 'SanPedro' , 57,-24 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (70,19, 'Quito' , 78,0 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (71,19, 'Guayaquil' , 80,-2 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (72,20, 'Merida' , 89,21 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (73,20, 'Monterrey' , 100,26 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES (74,20, 'Guadalajara' , 103,21 );
/*****/

```