

Relación Maestro–Detalle con Zeos, Firebird y Lazarus o Delphi

Esto es un tutorial sencillo que permite crear una relación Maestro – Detalle entre 2 tablas de una base de datos usando los componentes Zeos (6.6.6) y Firebird (2.1.3).

Está hecho en Lazarus (0.9.31), utilizando la distribución CodeTyphon (1.4) de Lazarus (que incluye los componentes Zeos) y en Windows XP.

También debería funcionar bien con Delphi, en Lazarus en Linux y con otras versiones de Firebird, aunque no se ha probado.

En este manual se detallarán todos los pasos a realizar. Es importante ir realizando el proceso paso por paso.

Fue realizado en Lazarus en español, por ello es posible que alguna imagen o el nombre de algún menú no sea igual al de Lazarus en otro idioma (como el inglés).

Crear una nueva aplicación.
Menú Archivo/Nuevo y seleccionar Aplicación.

Grabar esa aplicación en una carpeta.
Menú Archivo/Guardar todo...

Grabar un archivo en la misma carpeta en la que se guardó la aplicación en el paso anterior. Debe llamarse '**create.sql**'.

Para ello puede usar 'Notepad', 'Bloc de notas' o cualquier otro editor de texto.

Ese archivo incluirá las órdenes SQL que crearán las tablas y los registros de la base de datos de ejemplo que se usará en este manual.

El texto que debe contener este archivo está al final de este manual.

Seleccionar la pestaña de componentes Zeos (Zeos Access) en la paleta de componentes.



Añadir al Form un componente TZconnection (**Zconnection1**) y un componente TZSQLProcessor (**ZSQLProcessor1**).

En el componente ZSQLProcessor1, en la propiedad **Connection**, poner Zconnection1.

Añadir un componente Tbutton (**Button1**), de la pestaña Standard.

En su propiedad **Caption**, poner el texto '&Create DB / Crear BD' (sin comillas).

Hacer doble clic sobre el componente. Cuando aparezca el código para el evento Button1Click (Procedimiento **Tform1.Button1Click**), añadir este código:

```
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;

```

Añadir un componente TButton (**Button2**), de la pestaña Standard.
 En su propiedad **Caption**, poner el texto '&Exit / Salir' (sin comillas).
 Hacer doble clic sobre el componente en el Form. Cuando aparezca el código para el evento Button2Click (Procedimiento **Tform1.Button2Click**), añadir este código:

```
Close ;
```

Hacer clic sobre el Form (normalmente sería Form1) y, en el inspector de objetos, hacer clic en la pestaña Eventos.

Luego, en el evento **OnCreate**, hacer clic en el botón con tres puntos '...' y añadir este código:

```

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

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

```

En la sección 'Var' (antes de 'implementation'), añadir este código para declarar la variable aa como pública:

```
aa : string ;
```

Guardar los cambios hechos en la aplicación (**Ctrl+s**) y compilar (**F9**).

Cuando se muestre la aplicación, hacer clic en el botón 'Create DB / Crear BD'. Si todo va bien, aparecerá el mensaje indicando que la BD ha sido creada.

Se cierra la aplicación haciendo clic en el botón 'Salir'.

El archivo 'cities.fdb' se ha creado en la carpeta de la aplicación.

En el componente **ZConnection1**:

- En la propiedad **Database**, hacer clic en el botón con 3 puntos '...' y seleccionar la base de datos 'cities.fdb' creada en el paso anterior.
- En la propiedad **Password**, poner 'masterkey' (sin comillas).
- En la propiedad **Protocol**, seleccionar 'Firebird-2.1'. Se hace así porque estamos usando Firebird 2.1.3, si se usara otra versión de Firebird, habría que seleccionar el protocolo que le corresponda.
- En la propiedad **User**, poner 'sysdba' (sin comillas).

Seleccionar la pestaña de componentes Zeos.

Añadir al Form un componente **TZQuery** (**ZQuery1**) y un componente **TZUpdateSQL** (**ZUpdateSQL1**).

En el componente **Zquery1**:

- en la propiedad **Connection**, poner **Zconnection1**.
- en la propiedad **UpdateObject**, poner **ZUpdateSQL1**.
- en la propiedad **SQL**, hacer clic en el botón con 3 puntos '...' y escribir el siguiente código: 'Select * from COUNTRY' (sin comillas).

Hacer clic con el botón derecho del ratón sobre el componente **Zquery1** y seleccionar la primera opción del menú ('Editar campos...'):



Aparecerá una ventana vacía. Hacer clic en el botón con un signo '+'.
Aparecerá una ventana con los nombres de los campos de la tabla COUNTRY. Seleccionar todos los campos y hacer clic en el botón 'Create'.

Se cerrará la ventana y aparecerán los nombres de los campos en la ventana que inicialmente estaba vacía:



Cerrar esa ventana.

Hacer clic con el botón derecho del ratón sobre el componente **ZUpdateSQL1** y seleccionar la primera opción del menú ('UpdateSql editor...'):



Aparecerá una ventana. En ella, hacer clic en el campo COUNTRY_ID de 'Key fields:', para que quede como se ve en la imagen:

Luego, hacer clic en el botón 'Generate SQL'.

Se activará la pestaña 'SQL' y mostrará el código generado automáticamente.

Este código se modificará para 'Modify' y para 'Delete'.

En el código de **Modify**, el código final será:

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

En el código de **Delete**, el código final será:

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

Cuando se hayan hecho las modificaciones, hacer clic en el botón 'OK'.

Añadir un componente TDatasource (**Datasource1**) de la pestaña Data Access.
En su propiedad **DataSet** escribir Zquery1.

Seleccionar la pestaña de componentes Zeos.

Añadir al Form un componente TZQuery (**ZQuery2**) y un componente TZUpdateSQL (**ZUpdateSQL2**).

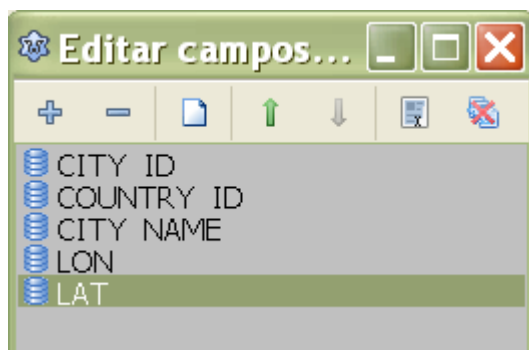
En el componente Zquery2:

- en la propiedad **Connection**, poner Zconnection1.
- en la propiedad **DataSource**, poner Datasource1.
- en la propiedad **UpdateObject**, poner ZUpdateSQL2.
- en la propiedad **SQL**, hacer clic en el botón con 3 puntos '...' y escribir el siguiente código:
'Select * from CITY where COUNTRY_ID=:COUNTRY_ID' (sin comillas).

Hacer clic con el botón derecho del ratón sobre el componente **Zquery2** y seleccionar la primera opción del menú ('Editar campos...'):

Aparecerá una ventana vacía. Hacer clic en el botón con un signo '+'.
Aparecerá una ventana con los nombres de los campos de la tabla CITY. Seleccionar todos los campos y hacer clic en el botón 'Create'.

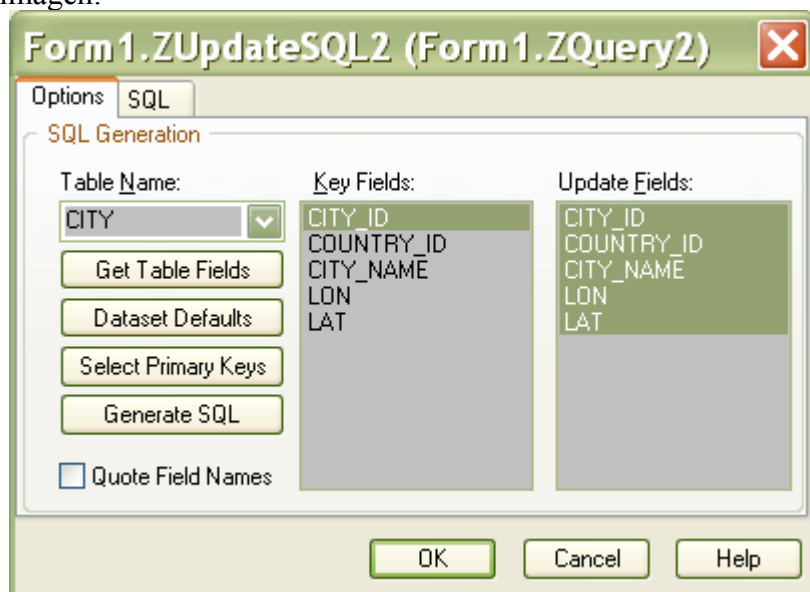
Se cerrará la ventana y aparecerán los nombres de los campos en la ventana que inicialmente estaba vacía:



Cerrar esa ventana.

Hacer clic con el botón derecho del ratón sobre el componente **ZUpdateSQL2** y seleccionar la primera opción del menú ('UpdateSql editor...'):

Aparecerá una ventana. En ella, hacer clic en el campo CITY_ID de 'Key fields:', para que quede como se ve en la imagen:



Luego, hacer clic en el botón 'Generate SQL'.

Se activará la pestaña 'SQL' y mostrará el código generado automáticamente.
Este código se modificará para 'Modify' y para 'Delete'.
En el código de **Modify**, el código final será:

```
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
```

En el código de **Delete**, el código final será:

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

Cuando se hayan hecho las modificaciones, hacer clic en el botón 'OK'

Añadir un componente TDataSource (**Datasource2**) de la pestaña Data Access.
En su propiedad **DataSet** escribir Zquery2.

Añadir un componente TDBGrid (**DBGrid1**) y un componente TDBNavigator (**DBNavigator1**) de la pestaña Data Controls.
En la propiedad **DataSource** de estos dos componentes, escribir Datasource1.

Añadir un componente TDBGrid (**DBGrid2**) y un componente TDBNavigator (**DBNavigator2**) de la pestaña Data Controls.
En la propiedad **DataSource** de estos dos componentes, escribir Datasource2.
En el componente DBNavigator2, en la propiedad **Visible**, escribir False.

Los comentarios son líneas de código que comienzan por los caracteres '//'.
En el procedimiento **TForm1.FormCreate** (donde anteriormente se añadió un código con comentarios), quitar los caracteres '/' en los comentarios de las últimas líneas del código.
Puede hacerlo a mano o puede seleccionar el texto de ese código y, en el menú Editar, hacer clic en 'Descomentar selección'.

Hacer clic sobre el **Form** y, en el inspector de objetos, hacer clic en la pestaña Eventos.
Luego, en el evento **OnDestroy**, hacer clic en el botón con tres puntos '...' y añadir este código:

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

Seleccionar **DBGrid1** haciendo clic sobre él y luego, manteniendo la tecla 'Shift' pulsada, hacer clic sobre DBGrid2. Así ambos componentes quedarán seleccionados.
Pulsar F11 y, en el Inspector de Objetos, hacer clic en la pestaña Eventos.
Luego, en el evento **OnEnter**, hacer clic en el botón con tres puntos '...' y añadir este código:

```

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

```

Hacer clic sobre **Zquery1** y, en el Inspector de Objetos, hacer clic en la pestaña Eventos. Luego, en el evento **BeforeDelete**, hacer clic en el botón con tres puntos '...' y añadir este código:

```

// 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 ;

```

En el procedimiento **TForm1.Button1Click** (donde anteriormente se añadió un código con comentarios), quitar los caracteres `//` en los comentarios de las 4 últimas líneas del código.

Guarde los cambios realizados (**Ctrl+s**) y compile la aplicación (**F9** ó **Ctrl+F9**).

En esta aplicación, al hacer clic sobre el botón 'Create DB / Crear BD', se borra la anterior BD 'cities.fdb' (si existía) y se crea una nueva BD con los mismos datos iniciales. Por ello, se puede borrar y modificar datos haciendo pruebas y, cuando se necesiten datos nuevos, se puede crear otra vez la BD inicial.

En la tabla Detalle (CITY), se puede borrar cualquier registro y modificar cualquier campo. Si se modifica la información del campo COUNTRY_ID, el resultado final debe ser un valor que exista en la tabla Maestro; en caso contrario, aparecerá un error.

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**.

Este manual está hecho para novatos y/o principiantes. El autor también es un novato y principiante. Por ello, es muy posible que existan pequeños, medianos y/o grandes errores. Aún así, se espera que sea útil.

Para sugerencias, correcciones, mejoras, etc., escribir a: **paratodo99@gmail.com**

En la dirección www.paratodo99.000a.biz/pas podrá descargar el código completo de esta aplicación.

Gracias por tu interés.

Este es el contenido del archivo '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, 'Gurie' , -52,47 );
INSERT INTO CITY (CITY_ID, COUNTRY_ID, CITY_NAME, LON, LAT ) VALUES ( 14,4, 'Sonali' , -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, 'Surabaja' , -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 );

```

/*****