Teoría Sqlite

SqLite 🡪 Es para crear y administrar bases de datos embebidas

ExecuteNonquery 🡪 Insert,update o Delete (valor lógico)

False (0) 🡪 no se realizó la tarea

True (1 o más) 🡪 Realizo la tarea

Una operación de escritura (Write)

Executequery 🡪 Select

DataSet o DataTable o Table

Cursor 🡪 SqLite

Una operación de lectura (Read)

Movimiento de la información en SqLite



Código clase ClsOpenHelper 🡪 clase que hereda de una clase ya creada en Android Studio para el manejo del SqLite

Código ClsOpenHelper

public class ClsOpenHelper extends SQLiteOpenHelper {  
  
 public ClsOpenHelper(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {  
 super(context, name, factory, version);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("create table TblCliente(Identificacion text primary key," +  
 "nombre text not null,correo text not null," +  
 "activo text not null default 'Si')");  
 db.execSQL("create table TblVehiculo(Placa text primary key," +  
 "marca text not null,modelo text not null," +  
 "activo text not null default 'Si')");  
 db.execSQL("create table TblVenta(codigo text primary key," +  
 "fecha text not null,Identificacion text not null," +  
 "Placa text not null,activo text default 'Si'," +  
 "constraint pk\_venta foreign key (Identificacion) " +  
 "references TblCliente(Identificacion)," +  
 "foreign key (Placa) references TbLVehiculo(Placa))");  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE TblCliente");  
 db.execSQL("DROP TABLE TblVehiculo");  
 db.execSQL("DROP TABLE TblVenta");{  
 onCreate(db);  
 }

}  
}

Código Xml de MainActivity

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:orientation="vertical"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:layout\_width="360dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="250dp"  
 android:layout\_gravity="center"  
 android:textSize="40dp"  
 android:onClick="Clientes"  
 android:text="Clientes" />  
  
 <Button  
 android:layout\_width="360dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_gravity="center"  
 android:onClick="Vehiculo"  
 android:textSize="40dp"  
 android:text="Vehiculo" />  
  
 <Button  
 android:layout\_width="360dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="40dp"  
 android:onClick="Ventas"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="Ventas" />  
</LinearLayout>

Código Java MainActivity

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 }  
  
 public void Clientes(View view){  
 Intent intclientes=new Intent(this,ClientesActivity.class);  
 startActivity(intclientes);  
 }  
  
 public void Vehiculo(View view){  
 Intent intvehiculo=new Intent(this,VehiculoActivity.class);  
 startActivity(intvehiculo);  
 }  
  
 public void Ventas(View view){  
 Intent intventas=new Intent(this,VentaActivity.class);  
 startActivity(intventas);  
 }  
}

Clase 16 02

Codigo Xml Clientes

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:orientation="vertical"  
 android:layout\_height="match\_parent"  
 tools:context=".ClientesActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="80dp"  
 android:gravity="center"  
 android:textSize="35dp"  
 android:text="Clientes" />  
  
 <EditText  
 android:id="@+id/etidentificacion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:layout\_marginTop="15dp"  
 android:textSize="35dp"  
 android:gravity="center"  
 android:hint="Digite identificacion"  
 android:inputType="number" />  
  
 <EditText  
 android:id="@+id/etnombre"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:textSize="35dp"  
 android:gravity="center"  
 android:layout\_marginTop="15dp"  
 android:hint="Digite nombre"/>  
  
 <EditText  
 android:id="@+id/etcorreo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:textSize="35dp"  
 android:gravity="center"  
 android:layout\_marginTop="15dp"  
 android:hint="Digite correo"  
 android:inputType="textEmailAddress" />  
  
 <CheckBox  
 android:id="@+id/cbactivo"  
 android:layout\_width="380dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginLeft="15dp"  
 android:textSize="35dp"  
 android:text=" Activo" />  
  
 <LinearLayout  
 android:layout\_width="380dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginLeft="15dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/btguardar"  
 android:layout\_width="174dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="Guardar"  
 android:textSize="25dp" />  
  
 <Button  
 android:id="@+id/btconsultar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="60dp"  
 android:textSize="25dp"  
 android:layout\_weight="1"  
 android:layout\_marginLeft="5dp"  
 android:text="Consultar" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="380dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginLeft="15dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/btanular"  
 android:layout\_width="174dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="Anular"  
 android:textSize="25dp" />  
  
 <Button  
 android:id="@+id/btcancelar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="60dp"  
 android:textSize="30dp"  
 android:layout\_weight="1"  
 android:layout\_marginLeft="5dp"  
 android:text="Cancelar" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/btregresar"  
 android:layout\_width="380dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="15dp"  
 android:layout\_gravity="center"  
 android:textSize="30dp"  
 android:text="Regresar" />  
  
</LinearLayout>

Sintaxis del código para consultar

“select \* from TblCliente where identificacion='"+identificacion+"'"

Apostrofe ( ‘ ) 🡪 Se coloca si la información es alfanumerica

Código Java de Clientes

public class ClientesActivity extends AppCompatActivity {  
  
 ClsOpenHelper admin=new ClsOpenHelper(this,"Concesionario.db",null,1);  
 EditText jetidentificacion,jetnombre,jetcorreo;  
 CheckBox jcbactivo;  
 String identificacion,nombre,correo;  
 long respuesta;  
 byte sw;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_clientes*);  
 getSupportActionBar().hide();  
 jetidentificacion=findViewById(R.id.*etidentificacion*);  
 jetnombre=findViewById(R.id.*etnombre*);  
 jetcorreo=findViewById(R.id.*etcorreo*);  
 jcbactivo=findViewById(R.id.*cbactivo*);  
 jetidentificacion.requestFocus();  
 sw=0;  
 }  
  
 public void Guardar(View view){  
 identificacion=jetidentificacion.getText().toString();  
 nombre=jetnombre.getText().toString();  
 correo=jetcorreo.getText().toString();  
 if (identificacion.isEmpty() || nombre.isEmpty() || correo.isEmpty()){  
 Toast.*makeText*(this, "Los datos son requeridos", Toast.*LENGTH\_SHORT*).show();  
 jetidentificacion.requestFocus();  
 }else{  
 SQLiteDatabase db=admin.getWritableDatabase();  
 ContentValues registro=new ContentValues();  
 registro.put("Identificacion",identificacion);  
 registro.put("nombre",nombre);  
 registro.put("correo",correo);  
 if (sw == 0)  
 respuesta=db.insert("TblCliente",null,registro);  
 else {  
 respuesta = db.update("TblCliente", registro, "identificacion='" + identificacion + "'", null);  
 sw=0;  
 }  
 if (respuesta > 0){  
 Toast.*makeText*(this, "Registro guardado", Toast.*LENGTH\_SHORT*).show();  
 Limpiar\_campos();  
 }else{  
 Toast.*makeText*(this, "Error guardando registro", Toast.*LENGTH\_SHORT*).show();  
 }  
 db.close();  
 }  
 }//Metodo guardar  
  
 public void Consultar(View view) {  
 identificacion=jetidentificacion.getText().toString();  
 if (identificacion.isEmpty()){  
 Toast.*makeText*(this, "Identificacion requerida para consultar", Toast.*LENGTH\_SHORT*).show();  
 jetidentificacion.requestFocus();  
 }else{  
 SQLiteDatabase db=admin.getReadableDatabase();  
 Cursor fila=db.rawQuery("select \* from TblCliente where identificacion='"+identificacion+"'",null);  
 if (fila.moveToNext()){  
 sw=1;  
 jetnombre.setText(fila.getString(1));  
 jetcorreo.setText(fila.getString(2));  
 if (fila.getString(3).equals("Si"))  
 jcbactivo.setChecked(true);  
 else  
 jcbactivo.setChecked(false);  
 }else{  
 Toast.*makeText*(this, "Registro no existe", Toast.*LENGTH\_SHORT*).show();  
 }  
  
  
 db.close();  
 }  
 }  
 private void Limpiar\_campos(){  
 jetcorreo.setText("");  
 jetidentificacion.setText("");  
 jetnombre.setText("");  
 jcbactivo.setChecked(false);  
 jetidentificacion.requestFocus();  
 sw=0;  
 }  
}