

Aula Prática 4

SQLite

SQLite é um Banco de Dados Open Source que já vem embutido no Sistema Android. É um Banco de Dados relacional suportando a sintaxe SQL, operações e instruções, requerendo pouca memória em tempo de execução. Sua estrutura está contida em apenas em um único arquivo no sistema e o acesso aos dados é implementado por uma biblioteca de funções escritas em C.

É um gerenciador de banco de dados “auto-suficiente”, ou seja, independente de uma estrutura cliente-servidor. Neste caso ele necessita de um apoio mínimo de bibliotecas externas ou até mesmo do Sistema Operacional, se tornando muito fácil para ser adaptado para o uso de dispositivos embarcados. Isto o torna flexível para muitas plataformas.

É muito utilizado em Dispositivos e sistemas embarcados (telefone celular, PDAs, MP3, entre outros), Aplicações Desktop, Sites com poucos acessos e também pode ser útil para o aprendizado.

Tipos de Campos

Na realidade no SQLite, o tipo de dado de um valor está associado com o valor propriamente dito, diferente de outros Bancos de Dados, se tornando um sistema de tipos dinâmicos.

Um campo de uma tabela em SQLite pode receber qualquer tipo de dado, ignorando o tipo informado no comando CREATE TABLE.

Em poucas palavras, podemos dizer que no SQLite existem classes de armazenamento, veja a seguir para maiores detalhes.

- **NULL**: O campo não deve ser Nulo, característica encontrada como em qualquer outro Banco de Dados.
- **INTEGER**: inteiro com sinal, armazenado em 1, 2, 3, 4, 6 ou 8 bytes dependendo da grandeza do valor.
- **REAL**: valor de ponto flutuante armazenado em 8 bytes.
- **TEXT**: uma string armazenada usando UTF-8, UTF-16BE ou UTF-16LE.
- **BLOB**: armazena campos do tipo Blob.

Atividade 1:

- Crie um projeto padrão;
- Crie uma nova classe no projeto chamada **DataHelper**, e implemente o código conforme abaixo:

```
public class DataHelper {

    // Atributos
    private static final String DATABASE_NAME = "DB_AULA7_PDM.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NAME = "TB_CLIENTE";
    private Context context;
    private SQLiteDatabase db;
    private SQLiteStatement insertStmt;
    private static final String INSERT = "INSERT INTO " + TABLE_NAME + " (nom_cli) VALUES (?)";

    // Construtor da classe
    public DataHelper (Context context)
    {
        this.context = context;
        OpenHelper openHelper = new OpenHelper(this.context);
        this.db = openHelper.getWritableDatabase();
        this.insertStmt = this.db.compileStatement(INSERT);
    }
}
```

```

// Método de Insert no Banco de dados
public long Insert(String name)
{
    this.insertStmt.bindString(1, name);
    return this.insertStmt.executeInsert();
}

// Método para exclusão de todos os registros
public void DeleteAll()
{
    this.db.delete(TABLE_NAME, null, null);
}

// Retorna lista com todos os itens cadastrados no banco de dados
public List<String> SelectAll()
{
    List<String> list = new ArrayList<String>();

    Cursor cursor = this.db.query(TABLE_NAME, new String[] { "cod_cli", "nom_cli" }, null,
null, null, null, "cod_cli");

    if (cursor.moveToFirst()) {
        do {
            list.add(cursor.getString(0) + " - " + cursor.getString(1));
        } while (cursor.moveToNext());
    }

    if (cursor != null && !cursor.isClosed()) {
        cursor.close();
    }

    return list;
}

// Classe responsável pelo gerenciamento e criação do banco de dados
private static class OpenHelper extends SQLiteOpenHelper
{
    OpenHelper(Context context)
    {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + "(cod_cli INTEGER PRIMARY KEY, nom_cli
TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.w("Example", "Upgrading database, this will drop tables and recreate.");
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
}

```

- No arquivo que representa a View, activity_main, crie um campo de texto (nome), um botão de insert e um campo de apresentação de texto (TextView) na View do Projeto:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

    <TextView android:text="Nome:" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:id="@+id/nome"
        android:layout_below="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="false"
        android:layout_alignParentRight="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Insert"
    android:id="@+id/insert"
    android:layout_below="@+id/nome"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:id="@+id/listaNomes"
    android:layout_below="@+id/insert"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>

```

- Na Activity do projeto (MainActivity), implemente o código abaixo:

```

public class MainActivity extends AppCompatActivity {

    Button btnInsert;
    TextView listaNomes;
    DataHelper dB;
    EditText fieldName;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });

        // Novas instâncias dos elemntos da View
        dB = new DataHelper(this);
        btnInsert = (Button) findViewById(R.id.insert);
        listaNomes = (TextView) this.findViewById(R.id.listaNomes);
        fieldName = (EditText) findViewById(R.id.nome);

        // Ação do clique do botão INSERT
        btnInsert.setOnClickListener(new View.OnClickListener() {
            {
                @Override
                public void onClick(View v)
                {
                    // Inserção de registro na tabela
                    dB.Insert(fieldName.getText().toString());

                    // Lista de todos os nomes cadastrados na tabela
                    List<String> names = dB.SelectAll();

                    // Impressão dos itens cadastrados
                    StringBuilder sb = new StringBuilder();
                    sb.append("Nomes Cadastrados:\n");
                    for (String name : names) {
                        sb.append(name + "\n");
                    }
                    listaNomes.setText(sb.toString());
                }
            }
        });
    }
}

```

```

        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}

```

- Execute sua aplicação e teste o resultado!!!

Atividade 2:

- Baseado no exemplo da Atividade 1, complemente o projeto com as funcionalidades de **INSERT**, **DELETE** e **UPDATE**. Se for preciso crie novas telas e botões para realizar o controle de cada operação.