

Sqlite Android



Professor Msc Romulo Beninca

Objetivos da aula

Compreender como podemos utilizar banco de dados em dispositivos moveis.

Adicionando suporte a banco SQLite a nossa aplicação

Para fazer uso do SQLite no Android, o primeiro passo é a instanciação de um objeto do tipo SQLiteDatabase, sendo este o objeto que fará a comunicação com o banco de dados, permitindo a execução de comandos, como, por exemplo, a criação de tabelas, manipulação e consulta de registros. Mas antes para podermos chamar estas funções temos que incluir a biblioteca SQLitedatabase;

```
import android.database.sqlite.SQLiteDatabase;
```



Como manipular banco SQLite no android

Um objeto SQLite database pode ser inicializado com o comando

```
SQLITEDATABASE banco = openOrCreateDatabase("banco", getBaseContext().MODE_PRIVATE, null);
```

Para executar uma SQL podemos fazer

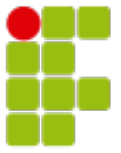
```
banco.execSQL("CREATE TABLE ...| INSERT ....");
```

Para uma recuperação de dados (SELECT)

```
ContentValues contentValues = new ContentValues();  
contentValues.put("nome", f.getNome());  
contentValues.put("tipo", f.getTipo());  
banco.insert("alunos", null, contentValues );
```

Para uma recuperação de dados (SELECT)

```
Cursor cursor = banco.rawQuery("SELECT id, nome FROM alunos ", null);  
cursor.moveToFirst();
```



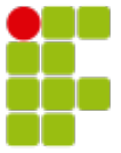
Definido um banco de dados SQLite para aplicação

Para utilizarmos um banco de dados SQLite em uma aplicação android basta definirmos uma propriedade SQLiteDatabase.

Trecho de código da classe ActivityMain.java

```
import android.database.sqlite.SQLiteDatabase;
public class MainActivity extends AppCompatActivity {
    private SQLiteDatabase banco;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void insereAluno() {
        ...
    }
    public void showAlunos() {
        ...
    }
}
```



Definindo DDL de criação do banco de dados.

Utilizando o método **openOrCreateDatabase** conseguimos solicitar a abertura de um banco de dados, ou criação caso este não exista.

Já o método **exeSQL** executa uma SQL qualquer.

```
public class MainActivity extends AppCompatActivity {
    private SQLiteDatabase banco;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        banco = this.openOrCreateDatabase("banco", getBaseContext().MODE_PRIVATE, null);
        banco.execSQL("CREATE TABLE IF NOT EXISTS alunos (_id INTEGER PRIMARY KEY AUTOINCREMENT, nome TEXT )" );
    }
    public void insereAluno() {
        ...
    }
    public void showAlunos() {
        ...
    }
}
```



Definindo DDL de criação do banco de dados.

Utilizando o método **openOrCreateDatabase** conseguimos solicitar a abertura de um banco de dados, ou criação caso este não exista.

Já o método **exeSQL** executa uma SQL qualquer.

```
public class MainActivity extends AppCompatActivity {
    private SQLiteDatabase banco;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        banco = this.openOrCreateDatabase("banco", getBaseContext().MODE_PRIVATE, null);
        banco.execSQL("CREATE TABLE IF NOT EXISTS alunos (_id INTEGER PRIMARY KEY AUTOINCREMENT, nome TEXT)");
    }
    public void insereAluno() {
        ...
    }
    public void showAlunos() {
        ...
    }
}
```



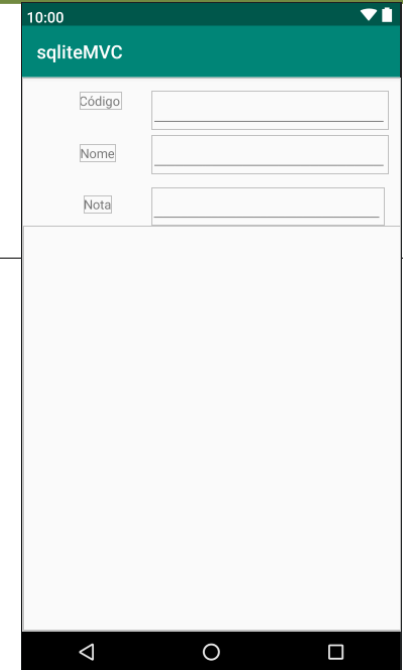
Definindo DDL de criação do banco de dados.

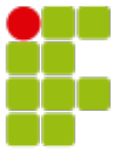
Para inserir um registro o código abaixo adicionamos dois campos EditText para que o usuário informe os dados.

```
public class MainActivity extends AppCompatActivity {
    private SQLiteDatabase banco;
    private EditText etCodigo,etNomeUC,etNota;
    private ListView listView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvNome= (TextView) findViewById(R.id.editTextCodigo);
        etNome= (EditText) findViewById(R.id.editTextNome);

        banco = this.openOrCreateDatabase("banco", getBaseContext().MODE_PRIVATE,null);
        banco.execSQL("CREATE TABLE IF NOT EXISTS alunos (_id INTEGER PRIMARY KEY AUTOINCREMENT,nome TEXT )" );
    }
    public void insereNota() {
        ContentValues contentValues = new ContentValues();
        contentValues.put("nome",f.getNome());
        this.banco.insert("alunos",null,contentValues );
    }
    public void showAlunos() {
        ...
    }
}
```





Definindo DDL de criação do banco de dados.

Para inserir um registro na tabela aluno podemos utilizar o comando insert, que recebe como parâmetros a string tabela null e um ContentValues

```
public class MainActivity extends AppCompatActivity {
    private SQLiteDatabase banco;
    private EditText etCodigo,etNome;
    private ListView listView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        etCodigo= (EditText) findViewById(R.id.editTextCodigo);
        etNome= (EditText) findViewById(R.id.editTextNome);

        banco = this.openOrCreateDatabase("banco", getBaseContext().MODE_PRIVATE,null);
        banco.execSQL("CREATE TABLE IF NOT EXISTS alunos (_id INTEGER PRIMARY KEY AUTOINCREMENT,nome TEXT )" );
    }
    public void insereAluno() {
        ContentValues contentValues = new ContentValues();
        contentValues.put("nome",etNome.getText().toString());
        this.banco.insert("alunos",null,contentValues );
    }
    public void showAlunos() {
        ...
    }
}
```

Suporte a SQLite no Android

Recuperando resultado de uma SQL

Como já fora dito anteriormente objeto da classe Cursor é uma tabela de dados, que armazena os registros recuperados pela SQL executada.

Esta tabela possui um ponteiro que indica qual é a linha de registro que será lida quando solicitarmos os dados recuperados. No momento em que o curso é recuperado ele aponta para ultima linha, para movermos este ponteiro para primeira linha dos registros recuperados executamos moveToFirst.

Trecho de código java para recuperar dados

```
Cursor cursor = banco.rawQuery("SELECT id, nome FROM alunos ", null);  
cursor.moveToFirst();
```

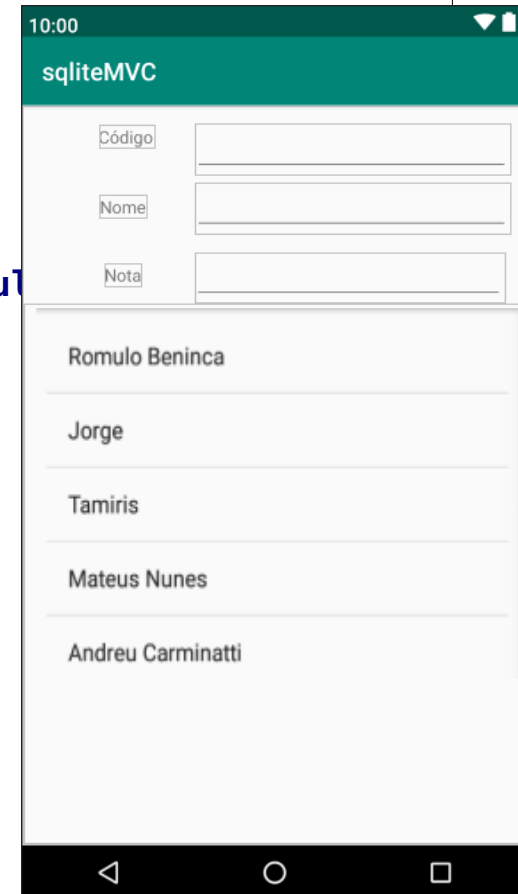




Definindo DDL de criação do banco de dados.

Utilizando o método **openOrCreateDatabase** conseguimos solicitar a abertura de um banco de dados, ou criação caso este não exista.

```
public class MainActivity extends AppCompatActivity {
    private SQLiteDatabase banco;
    private EditText etCodigo, etNome;
    private ListView listView;
    @Override
    protected void onCreate(Bundle savedInstanceState) { ... }
    public void insereNota() { ... }
    public void showAlunos() {
        Cursor cursor = banco.rawQuery("SELECT id, nome FROM alunos ", null);
        cursor.moveToFirst();
        ArrayList<String> itens = new ArrayList<>();
        do {
            String s = cursor.getString( cursor.getColumnIndex("nome"));
            itens.add(s);
        }while (cursor.moveToNext()) ;
        ListView list;
        list = findViewById(R.id.listView);
        ArrayAdapter<String> adapter = new ArrayAdapter<String> (
            getApplicationContext(),
            android.R.layout.simple_list_item_1,
            android.R.id.text1,
            itens);
        list.setAdapter(adapter);
    }
}
```



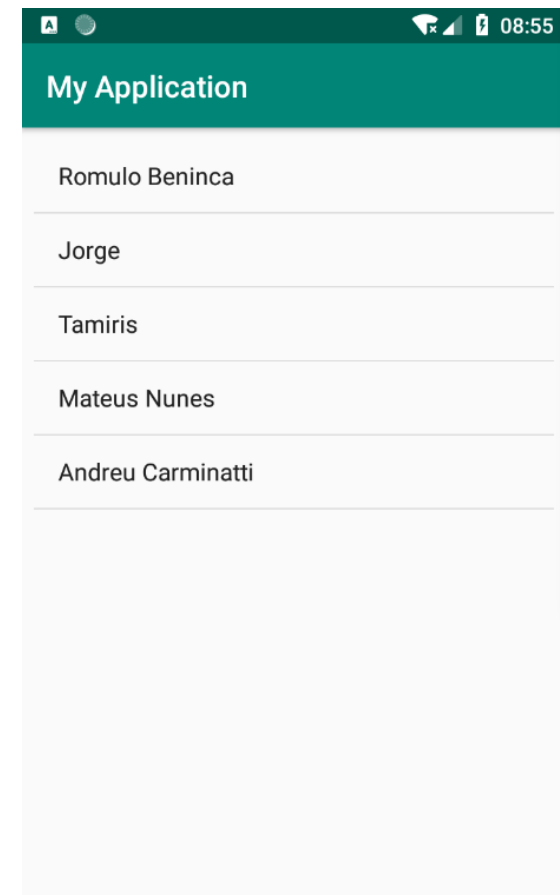
Suporte a SQLite no Android

Apresentado dados recuperados em um ListView

Por fim, para apresentarmos estes dados em um ListView podemos utilizar o código a baixo.

Trecho de código java para recuperar dados

```
Cursor cursor = banco.rawQuery("SELECT id, nome FROM alunos ",
null);
cursor.moveToFirst();
cursor.getBlob()
ArrayList<String> itens = new ArrayList<>();
do {
    String s = cursor.getString( cursor.getColumnIndex("nome"));
    Log.i(" Resultado Sql :",s );
    itens.add(s);
}while (cursor.moveToNext()) ;
ListView list;
list = findViewById(R.id.listView);
ArrayAdapter<String> adapter = new ArrayAdapter<String> (
    getApplicationContext(),
    android.R.layout.simple_list_item_1,
    android.R.id.text1,
    itens
);
list.setAdapter(adapter);
```



O fonte do projeto esta disponivel em :

<https://bitbucket.org/professorromulobeninca/aula-08-sql-raw/src>