



Desenvolvimento para Dispositivos Móveis

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

9. BANCO DE DADOS



SQL - *Structured Query Language*

- a) **DDL** (Data **D**efinition Language – Linguagem de **D**efinição de Dados)
- b) **DML** (Data **M**anipulation Language – Linguagem de **M**anipulação de Dados)
- c) **DCL** (Data **C**ontrol Language – Linguagem de **C**ontrole de Dados)

9. BANCO DE DADOS



Comandos DDL

SENTENÇA SQL	SINTAXE
CREATE DATABASE	CREATE DATABASE nome_bd
CREATE TABLE	CREATE TABLE nome_tabela (nome_coluna1 tipo_dado, nome_coluna2 tipo_dado, nome_coluna3 tipo_dado, ...)

9. BANCO DE DADOS



Comandos DDL

SENTENÇA SQL	SINTAXE
DROP DATABASE	DROP DATABASE nome_bd
DROP TABLE	DROP TABLE nome_tabela

9. BANCO DE DADOS



Comandos DDL

SENTENÇA SQL	SINTAXE
ALTER TABLE	ALTER TABLE nome_tabela ADD nome_coluna
	ALTER TABLE nome_tabela DROP COLUMN nome_coluna

9. BANCO DE DADOS



Comandos DML

SENTENÇA SQL	SINTAXE
SELECT	SELECT nome_coluna(s) FROM nome_tabela WHERE nome_coluna = / > / < / IN / LIKE ...
INSERT INTO	INSERT INTO nome_tabela VALUES (valor1, valor2, valor3, ...) ou INSERT INTO nome_tabela (coluna1, coluna2, coluna3, ...) VALUES (valor1, valor2, valor3, ...)

9. BANCO DE DADOS



Comandos DML

SENTENÇA SQL	SINTAXE
UPDATE	UPDATE nome_tabela SET coluna1 = valor, coluna2 = valor, ... WHERE alguma_coluna = algum_valor
DELETE	DELETE FROM nome_tabela WHERE alguma_coluna = algum_valor ou DELETE FROM nome_tabela (Obs: Deleta todos os registros da tabela)

9. BANCO DE DADOS



Comandos DCL

SENTENÇA SQL	SINTAXE
GRANT	Autorizar o usuário a executar ou setar operações GRANT nome_privilégio ON nome_objeto TO {nome_usuario PUBLIC nome_papel} [WITH GRANT OPTION];
REVOKE	Restringir ou remover acesso de executar operações REVOKE nome_privilégio ON nome_objeto FROM {nome_usuario PUBLIC nome_papel}

9. BANCO DE DADOS



Funções de Agregação

FUNÇÃO	DESCRIÇÃO
MIN()	Pega o valor mínimo de um campo/coluna
MAX()	Pega o valor máximo de um campo/coluna
COUNT()	Pega o número de registros de uma consulta (select)
SUM()	Pega a soma (<i>sum</i>) de todos os valores de um campo/coluna
AVG()	Pega a média (<i>average</i>) de todos os valores de um campo/coluna

9. BANCO DE DADOS



Banco de Dados - Mais comandos...

http://www.w3schools.com/sql/sql_quickref.asp
http://www.w3schools.com/sql/sql_join.asp



9. BANCO DE DADOS



- A imensa maioria dos aplicativos usa banco de dados
- **SGBD**: SQLite;
- Baseado em arquivos (extensão .db) --> não necessita de um processo servidor;
- Armazena as tabelas, views, índices e triggers no arquivo --> leitura e escrita;
- O BD é acessível por qualquer classe da aplicação que o criou. Mas não pode ser acessado por outro app;
- Quando o app é desinstalado, o BD também é removido.

9. BANCO DE DADOS



- Existem classes prontas programadas em Java que realizam todo o trabalho de se comunicar com o BD;
- Principais Classes:
 1. **SQLiteOpenHelper**: Classe responsável pela criação do banco, pelo versionamento e provê acesso de leitura ou de escrita no BD;
 2. **SQLiteDatabase**: Classe que contém os métodos de manipulação dos dados no banco (select, insert, update, delete).



9. BANCO DE DADOS



- As classes facilitam:
 - a criação,
 - o versionamento,
 - a manipulação dos dados do BD.
- Existem várias formas de fazer a programação do app com o Banco de Dados.
- Vamos ver uma delas... (7 passos para inserção)



9. BANCO DE DADOS – Passos para utilização de BD em um app Android (*Depois da criação do projeto...*)



- Passo 1: criar uma variável da classe **SQLiteDatabase** dentro da Activity:

```
private SQLiteDatabase banco;
```

- Passo 2: instanciar o objeto através do método **openOrCreateDatabase**:

```
banco = this.openOrCreateDatabase (
    "nome_banco", Context.MODE_PRIVATE, null );
```

9. BANCO DE DADOS - Passos para utilização de BD em um app Android



- Passo 3: criar a(s) tabela(s) no BD:

```
banco.execSQL( " CREATE TABLE IF NOT EXISTS
    nome_tabela (
        _id INTEGER PRIMARY KEY AUTOINCREMENT,
        coluna1 TEXT,
        coluna2 DECIMAL
    )"
);
```

9. BANCO DE DADOS - Passos para utilização de BD em um app Android



Dentro de um método dinâmico...

- Passo 4: criar objeto da classe **ContentValues**:

```
ContentValues registro = new ContentValues( );
```

- Passo 5: Usar o objeto do passo 4 para adicionar valores na tabela do BD através do método **put**:

```
registro.put( "nome_coluna1", valor_col_1 );
registro.put( "nome_coluna2", valor_col_2 );
```


9. BANCO DE DADOS - Passos para utilização de BD em um app Android



Dentro de um método dinâmico...

- Passo 6: inserir registro no banco de dados através do método **insert**, passando três argumentos:

```
banco.insert( "nome_tabela", null, registro );
```

- Passo 7: Mostrar um **Toast** de confirmação:

```
Toast.makeText( this ,  
                "Dado inserido!" ,  
                Toast.LENGTH_LONG ).show();
```

9. BANCO DE DADOS – Passos para utilização de BD em um app Android



- É uma boa prática de programação, fazer a comunicação com o BD através de uma estrutura **try-catch**, pois facilita o tratamento de erros (exceções):

```
try {  
    //LÓGICA DE MANIPULAÇÃO DE DADOS  
} catch(Exception e) {  
    Toast.makeText( this ,  
                    "Falha ao manipular BD" ,  
                    Toast.LENGTH_LONG ).show( );  
}
```

9. BANCO DE DADOS - Passos para utilização de BD em um app Android



• Outros métodos para realizar outras manipulações:

• **Atualização:**

```
banco.update( "nome_tabela", registro, "_id = " + VALOR_DO_ID, null );
```

• **Remoção:**

```
banco.delete( "nome_tabela", "_id = " + VALOR_DO_ID, null );
```

• **Inserção:**

```
banco.insert( "nome_tabela", null, registro );
```



Desenvolvimento para Dispositivos Móveis

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

ALERT DIALOG - clique no item da lista



1. Quando o usuário clicar em um item do **RecyclerView**, opções devem aparecer para manipulação e compartilhamento de dados.
2. As opções serão mostradas em um **AlertDialog** (figura).



AlertDialog

ALERT DIALOG - clique no item da lista



- Passos para criar o AlertDialog
1. Criar a lista de opções no [strings.xml](#). **<LAYOUT>**
 2. Dentro do Adapter, implementar a interface [View.OnClickListener](#) na classe estática (static) [ViewHolder](#), sobrescrever o método [onClick](#) e [atribuir o clique](#) no item. **<JAVA>**
 3. Dentro do método [onClick](#), pegar a cor clicada, criar um [builder](#) do AlertDialog, atribuir o título, as opções (XML) e o que vai acontecer quando cada item do AlertDialog for clicado. **<JAVA>**
 4. Fazer uma série de outras mudanças que serão mostradas pelo professor durante o desenvolvimento do projeto prático!

ALERT DIALOG - clique no item da lista



- No arquivo: **res/values/strings.xml**
- Criar um array de strings que vai servir para mostrar opções dentro do AlertDialog:

```
<string-array name="opcoes_menu_lista">
  <item>Compartilhar</item>
  <item>Editar</item>
  <item>Remover</item>
  <item>Cadastrar Novo</item>
</string-array>
```

ALERT DIALOG - clique no item da lista



```
public static class ViewHolder
  extends RecyclerView.ViewHolder
  implements View.OnClickListener{
  @Override
  public void onClick(View v) {
    Cor corClicada = listaCores.get( getAdapterPosition() );
    // continuar aqui...
  }
}
```

ALERT DIALOG - clique no item da lista



```

@Override
public void onClick(View v) {
    AlertDialog.Builder builder = new AlertDialog.Builder( v.getContext() );
    builder.setTitle( "Opções" );
    builder.setItems ( R.array.opcoes_menu_lista, new
        DialogInterface.OnClickListener() {
            public void onClick ( DialogInterface dialog, int opcaoMenu ) {
                //continua aqui...
            }
        }
    );
}

```

Clique no item do RecyclerView

Clique no item do AlertDialog

ALERT DIALOG - clique no item da lista



```

public void onClick ( DialogInterface dialog, int opcaoMenu ) {
    switch ( opcaoMenu ) {
        case 0: // Compartilhar
            // ENVIAR DADOS PELA INTENT IMPLÍCITA
            break;
        case 1: // Editar
            // IR PARA ACTIVITY DE INSERIR ITEM, LEVANDO O ID
            // DO ITEM CLICADO, FAZER UM SELECT E DEPOIS ATUALIZAR
            break;
        // continua...
    }
}

```

ALERT DIALOG - clique no item da lista



```
public void onClick ( DialogInterface dialog, int opcaoMenu ) {
    switch ( opcaoMenu ) {
        case 2: // Remover
            // FAZER CONFIRMAÇÃO DA AÇÃO ATRAVÉS DE OUTRO
            TIPO DE ALERT DIALOG
            // REMOVER DO BANCO DE DADOS E MONTAR NOVAMENTE
            A LISTA
            break;
        case 3: // Inserir novo
            // SIMPLEMENTE IR PARA A ACTIVITY DE INSERIR
            break;
        // continua...
    }
}
```

ALERT DIALOG - clique no item da lista



```
default:
    Toast.makeText( v.getContext(),
        "Erro! Opção inválida.",
        Toast.LENGTH_LONG).show();
    } // fim do switch
    } // fim do método onClick
}); // fim do método setItems
AlertDialog dialog = builder.create();
dialog.show();
} // fim do método onClick
```

ALERT DIALOG - clique no item da lista



- Na classe: Tela2
- Caso o usuário decida **COMPARTILHAR**:

case 0: //Compartilhar

```
String msgComp = "Selecionei a cor " + corClicada.getNome();
Intent comp = new Intent( Intent.ACTION_SEND );
comp.setType( "text/plain" );
comp.putExtra( Intent.EXTRA_SUBJECT, "Info Cor" );
comp.putExtra( Intent.EXTRA_TEXT, msgComp );
startActivity( Intent.createChooser( comp, "Compartilhar..." ) );
```

break;

ALERT DIALOG - clique no item da lista



- Na classe: Tela2
- Caso o usuário decida **EDITAR**:

case 1: //Editar

```
Intent editarTela1 = new Intent( Tela2.this, InserirCorActivity.class );
editarTela1.putExtra( "id_editar", corClicada.getId() );
startActivity( editarTela1 );
```

break;

NECESSÁRIO
AJUSTE NESSA
CLASSE PARA
EDITAR O ITEM

ALERT DIALOG - clique no item da lista



case 2: **//Remover**

```
try {
    banco.delete( CorBD.TabCor.TABELA,
        CorBD.TabCor.COL_ID + " = ?",
        new String[ ]{ corClicada.getId() + "" } );
    removerItemLista( getAdapterPosition() );
    Toast.makeText( Tela2.this, "Dado removido do banco de dados.",
        Toast.LENGTH_SHORT).show( );
} catch(Exception e) {
    Toast.makeText(Tela2.this, "Erro ao tentar remover: " +
        e.getMessage( ), Toast.LENGTH_SHORT).show( );
}
break;
```

ALERT DIALOG - clique no item da lista



Colocar esse método na classe CorAdapter:

```
public void removerItemLista(int posicao) {
    this.listaCores.remove(posicao);
    notifyItemRemoved(posicao);
    notifyItemRangeChanged(posicao, getItemCount() - posicao);
}
```


ALERT DIALOG - clique no item da lista



- Caso o usuário decida **INSERIR NOVO**:

case 3: // Inserir Novo

```
Intent telalns = new Intent(v.getContext(), InserirCorActivity.class);
v.getContext().startActivity(telalns);
```

break;

ALERT DIALOG - clique no item da lista



- Depois de fazer o método `onClick` na classe `ViewHolder`, é necessário atribuir essa ação ao método **`setOnClickListener`** do item da lista (comando na cor azul):

```
public ViewHolder(@NonNull View itemView) {
    super(itemView);
    txtCor = itemView.findViewById(R.id.txt_cor);
    itemView.setOnClickListener(this);
}
```

ALERT DIALOG - Exercício

