



Banco de Dados SQLite

Prof. Diogo Soares



Banco de Dados



- O Banco de Dados do Android é o SQLite
 - Suporta a comandos SQL, Leve
 - Dados são salvos em um único arquivo local
 - Já vem instalado
 - *Sem necessidade de usuário e senha. Entretanto, seus dados estão seguros, pois só serão acessíveis por sua app*
 - <http://www.sqlite.org/>

Banco de Dados



- ▣ Para utilizar o banco de dados no app, iremos:
 - ▣ Criar uma nova classe auxiliar para usar o banco de dados
 - *Database.java*
 - ▣ Modificar a classe `PasswordDAO` para acessar os dados do banco de dados
 - *Ao invés de usar o `ArrayList`, como feito nos slides anteriores*

Auxiliar de Banco de Dados



- ▣ Nome da classe: `Database`
- ▣ Esta classe estende a classe `SQLiteOpenHelper`
- ▣ Terá constantes e métodos para auxiliar o acesso ao BD
 - ▣ Explicados a seguir
- ▣ Ela será instanciada dentro de cada classe DAO
 - ▣ DAO: Data Access Objects

Auxiliar de Banco de Dados

▣ Constantes:

▣ DATABASE_VERSION

- *Versão do banco de dados. Incremente esse valor sempre que fizer modificação na estrutura do banco de dados ou quando quiser limpar o banco de dados (nessa implementação).*
- *Sempre que o android perceber que a versão dessa constante é maior que a última versão do BD no seu celular, ele irá executar o método `onUpdate`, mostrado no próximo slide*

▣ DATABASE_NAME

- *Nome do arquivo do banco de dados*

▣ SQL_CREATE_PASS, SQL_POPULATE_PASS, SQL_DELETE_PASS

- *Comandos SQL para criar, popular e remover a tabela de senhas*
- *Crie constantes diferentes para cada uma das tabelas do seu app*

Auxiliar de Banco de Dados

■ Métodos:

□ Database

- *Construtor, recebe o contexto (Activity) como parâmetro*
- *Executa o construtor da classe pai (SQLiteOpenHelper) passando o contexto, e o nome e a versão do banco de dados*

□ onCreate

- *Método executado quando o aplicativo é iniciado e nenhum banco de dados existe ainda*
- *Provavelmente o aplicativo acabou de ser instalado*

□ onUpgrade

- *Executado quando a versão do banco de dados do app atual é diferente da versão do banco de dados no arquivo do celular*
- *Provavelmente o aplicativo foi atualizado, será necessário executar comandos SQL para “converter” o BD do celular para a versão mais nova*
- *No exemplo a seguir, apagamos a tabela e criamos novamente*

Auxiliar de Banco de Dados

```
public class Database extends SQLiteOpenHelper {  
    public static final int DATABASE_VERSION = 4;  
    public static final String DATABASE_NAME = "PlainText.db";  
    private static final String SQL_CREATE_PASS = "CREATE TABLE passwords (" +  
        "id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, login TEXT," +  
        "password TEXT, notes TEXT)";  
    private static final String SQL_POPULATE_PASS = "INSERT INTO passwords VALUES " +  
        "(NULL, 'GMail', 'dovahkiin', 'Teste123', 'Nota de Teste')";  
    private static final String SQL_DELETE_PASS = "DROP TABLE IF EXISTS passwords";  
  
    public Database(Context context) {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    }  
  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL(SQL_CREATE_PASS);  
        db.execSQL(SQL_POPULATE_PASS);  
    }  
  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
        db.execSQL(SQL_DELETE_PASS);  
        onCreate(db);  
    }  
}
```

Versão do banco de dados

Comandos SQL para criar, popular e remover a tabela passwords

Executado quando o BD for criado pela primeira vez

Executado quando o BD for atualizado

Modificando a Classe PasswordDAO

- A classe PasswordDAO deixará de salvar os dados em um `ArrayList` e passará a usar o banco de dados
- Atributos e Construtor:

```
public class PasswordDAO {  
    private Context context;  
    private SQLiteDatabase database;  
  
    public PasswordDAO(Context context) {  
        this.context = context;  
        this.database = (new Database(context)).getWritableDatabase();  
    }  
}
```

Acessa o SQLite usando a
nossa classe auxiliar

Modificando a Classe PasswordDAO

■ Método getList

```
public ArrayList<Password> getList() {  
    ArrayList<Password> result = new ArrayList<Password>();  
    String sql = "SELECT * FROM passwords ORDER BY name";  
    Cursor cursor = database.rawQuery(sql, null);  
  
    while (cursor.moveToNext()) {  
        int id = cursor.getInt(0);  
        String name = cursor.getString(1);  
        String login = cursor.getString(2);  
        String password = cursor.getString(3);  
        String notes = cursor.getString(4);  
        result.add(new Password(id, name, login, password, notes));  
    }  
  
    return result;  
}
```

Retorna um ArrayList,
mas os dados virão do BD

Executa um comando SQL
no SQLite do Android

Para cada linha retornada
da consulta anterior ...

Acessa a primeira coluna
da linha como um inteiro

Cria um objeto Password
e insere na lista de retorno

Modificando a Classe PasswordDAO

■ Método add

```
public boolean add>Password password) {  
    String sql = "INSERT INTO passwords VALUES (NULL, "  
        + "'" + password.getName() + "', "  
        + "'" + password.getLogin() + "', "  
        + "'" + password.getPassword() + "', "  
        + "'" + password.getNotes() + "')";  
  
    try {  
        database.execSQL(sql);  
        Toast.makeText(context, "Senha salva!", Toast.LENGTH_SHORT).show();  
        return true;  
    }  
    catch (SQLException e) {  
        Toast.makeText(context, "Erro! " + e.getMessage(), Toast.LENGTH_SHORT).show();  
        return false;  
    }  
}
```

Adiciona um Password no banco de dados

Comando SQL para inserir os dados no BD

Executa o comando SQL

Erro ao executar o comando SQL

Modificando a Classe PasswordDAO

■ Método update

```
public boolean update>Password password) {  
    String sql = "UPDATE passwords SET "  
        + "name='" + password.getName() + "', "  
        + "login='" + password.getLogin() + "', "  
        + "password='" + password.getPassword() + "', "  
        + "notes='" + password.getNotes() + "' "  
        + "WHERE id=" + password.getId();  
  
    try {  
        database.execSQL(sql);  
        Toast.makeText(context, "Senha atualizada!", Toast.LENGTH_SHORT).show();  
        return true;  
    }  
    catch (SQLException e) {  
        Toast.makeText(context, "Erro! " + e.getMessage(), Toast.LENGTH_SHORT).show();  
        return false;  
    }  
}
```

Atualiza um Password no banco de dados

Comando SQL para atualizar os dados no BD

Executa o comando SQL

Erro ao executar o comando SQL

Modificando a Classe PasswordDAO

■ Método get

```
public Password get(int id) {  
    String sql = "SELECT * FROM passwords WHERE id=" + id;  
    Cursor cursor = database.rawQuery(sql, null);  
  
    if (cursor.moveToNext()) {  
        String name = cursor.getString(1);  
        String login = cursor.getString(2);  
        String password = cursor.getString(3);  
        String notes = cursor.getString(4);  
        return new Password(id, name, login, password, notes);  
    }  
  
    return null;  
}
```

Dado um ID, retorna o Password, com este ID

Executa um comando SQL no SQLite do Android

Acessa a primeira linha retornada da consulta

Acessa a segunda coluna da linha como uma string

Cria um objeto Password e o retorna

Se a senha com o ID passado não for encontrada, retorna nulo


Resultado


- ▣ Testando o app com acesso ao banco de dados

Testando o Banco de Dados

□ Resultado


PlainText


 **GMail**
dovahkiin >





PlainText


Senha


 Coursera

 sovngarde


 HunKaalZoor


 Notas






PlainText

 **Coursera**
sovngarde >

 **GMail**
dovahkiin >



Referências



- Tutorial da própria lib: [Salvar dados usando o SQLite | Desenvolvedores Android | Android Developers](#)
- Tutorial: [Utilizando Banco de Dados SQLite no Android | by Nelson Glauber | Medium](#)