

Banco de Dados

Bruno Lopes Alcantara Batista

Prazer em conhecer,

Bruno Lopes A Batista

Me. Ciência da Computação - UECE

Esp. em Desenvolvimento Web com JavaEE - FJN

Bel. em Sistemas de Informação - FJN

Professor Pesquisador - LARCES

Gerente de Projetos - LARCES

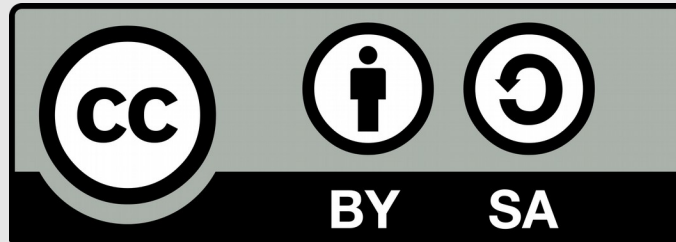
Gerente de Projetos - Athomustec

bruno@larces.uece.br

<http://www.larces.uece.br/bruno>



Copyleft



*Esta obra está licenciado com uma Licença
[Creative Commons Atribuição-CompartilhaIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)*

Agenda

- ☐ Introdução
- ☐ Utilizando o SQLite
- ☐ Let's try it!



Introdução

Introdução

- ❑ Salvar dados em um banco de dados é uma tarefa recorrente.
- ❑ O Android utiliza o SQLite como SGBD.
- ❑ Está disponível em todas as versões nativamente
- ❑ Utilizar o SQLite no Android é uma tarefa simples.

Utilizando o SQLite

Utilizando o SQLite

- ❑ Primeiro, deve-se criar uma classe Java que estende de SQLiteOpenHelper.
- ❑ Nessa classe deve-se, obrigatoriamente implementar os métodos onCreate() e onUpgrade()
- ❑ O método onCreate() é chamado quando o banco é criado pela primeira vez.
- ❑ O método onUpgrade() é chamado toda vez que a versão do banco de dados é modificada.
- ❑ Também existe o método onDowngrade() que retorna a base de dados para uma versão anterior e esse método é opcional.

Utilizando o SQLite

```
public class FeedReaderDbHelper extends SQLiteOpenHelper {
    // If you change the database schema, you must increment the database version.
    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "FeedReader.db";

    public FeedReaderDbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_ENTRIES);
    }
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // This database is only a cache for online data, so its upgrade policy is
        // to simply to discard the data and start over
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }
    public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        onUpgrade(db, oldVersion, newVersion);
    }
}
```

Utilizando o SQLite

- Depois de criada a classe Helper, basta inicia-la da seguinte maneira:

```
FeedReaderDbHelper mDbHelper = new FeedReaderDbHelper(getApplicationContext());
```

Utilizando o SQLite

Inserindo informações na base de dados

- Para inserir informações na base de dados, precisamos passar um objeto ContentValues para o método insert()

```
// Gets the data repository in write mode
SQLiteDatabase db = mDbHelper.getWritableDatabase();

// Create a new map of values, where column names are the keys
ContentValues values = new ContentValues();
values.put(FeedEntry.COLUMN_NAME_ENTRY_ID, id);
values.put(FeedEntry.COLUMN_NAME_TITLE, title);
values.put(FeedEntry.COLUMN_NAME_CONTENT, content);

// Insert the new row, returning the primary key value of the new row
long newRowId;
newRowId = db.insert(
    FeedEntry.TABLE_NAME,
    FeedEntry.COLUMN_NAME_NULLABLE,
    values);
```

Utilizando o SQLite

Lendo informações da base de dados

- Para ler informações da base de dados, utilizamos o método `query()` passando o critério de seleção e as colunas desejadas:

```
SQLiteDatabase db = mDbHelper.getReadableDatabase();

// Define a projection that specifies which columns from the database
// you will actually use after this query.
String[] projection = {
    FeedEntry._ID,
    FeedEntry.COLUMN_NAME_TITLE,
    FeedEntry.COLUMN_NAME_UPDATED,
    ...
};

// How you want the results sorted in the resulting Cursor
String sortOrder =
    FeedEntry.COLUMN_NAME_UPDATED + " DESC";

Cursor c = db.query(
    FeedEntry.TABLE_NAME, // The table to query
    projection,            // The columns to return
    selection,             // The columns for the WHERE clause
    selectionArgs,        // The values for the WHERE clause
    null,                 // don't group the rows
    null,                 // don't filter by row groups
    sortOrder,            // The sort order
);
```

Utilizando o SQLite

Excluindo informações do banco de dados

- Para excluir informações do banco de dados, devemos passar os critérios de seleção que definem as linhas a serem excluídas:

```
// Define 'where' part of query.  
String selection = FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";  
// Specify arguments in placeholder order.  
String[] selectionArgs = { String.valueOf(rowId) };  
// Issue SQL statement.  
db.delete(table_name, selection, selectionArgs);
```

Utilizando o SQLite

Atualizando informações do banco de dados

- Para atualizar um subconjunto de linhas da base de dados utilizamos o método `update()`.

```
SQLiteDatabase db = mDbHelper.getReadableDatabase();

// New value for one column
ContentValues values = new ContentValues();
values.put(FeedEntry.COLUMN_NAME_TITLE, title);

// Which row to update, based on the ID
String selection = FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";
String[] selectionArgs = { String.valueOf(rowId) };

int count = db.update(
    FeedReaderDbHelper.FeedEntry.TABLE_NAME,
    values,
    selection,
    selectionArgs);
```



Let's try it!

Let's try it!

- ☐ Implementar um CRUD de Person na aplicação Demo APP.

Obrigado!

