

# 1 Datenbank

In der Datenbank werden alle für die Applikation essentiellen Daten gespeichert.

## 1.1 SQLite

SQLite ist eine open-source library, sie implementiert ein unabhängiges, serverlos, zeroconf und transaktionales SQL-database-engine [dbSQLite [dbSQLite]]

## 1.2 Zugriff auf die Daten

Der Zugriff auf die abgespeicherten Daten erfolgt über 3 Schichten:

### 1.2.1 Physische Ebene

Alle Daten werden im Speicher des Mobilgerätes abgelegt und persistiert. Die Speicherung übernimmt das DBMS von SQLite.

### 1.2.2 DB Helper

Der DB Helper stellt die Datenbank als Objekt zur Verfügung, welches in den Repositories instanziiert wird.

### 1.2.3 Repositories

Die Repositories dienen als Puffer zwischen dem Datenbankobjekt und der Businesslogik. Jedes Repository ist als Singleton implementiert und um darauf zugreifen zu können muss es vorerst im Code instanziiert werden.

```
1 Repository repository = Repository.getInstance();
```

In den Repositories muss das Datenbankobjekt mithilfe des DB Helpers instanziiert werden ...

```
1 SQLiteDatabase db = getWritableDatabase();
2
3 private SQLiteDatabase getWritableDatabase() {
4     if (repoContext == null)
5         throw new IllegalStateException();
6
7     return FiplyDBHelper.getInstance(repoContext)
8         .getWritableDatabase();
9 }
```

... um über dieses Objekt dann mithilfe von SQL-Statements auf die Daten zugreifen bzw. die Daten manipulieren zu können.

```
1 SQLiteDatabase db = getWritableDatabase();
2
3 return db.query("SQL-STATEMENT");
```

Folgende Repositories sind in der Applikation vorhanden:

- Instruktionen-Repository
- Key-Value-Repository
- Phasen-Repository
- Plan-Repository
- Playlis-Songs-Repository
- Statistic-Repository
- Uebungen-Repository

## 1.3 Contract

Der Contract ist eine Datei in welcher die Metadaten der Datenbank zu finden sind. Dieser "Vertrag" existiert um den makellosen Zugriff auf die Datenbank sicherzustellen. Für jede Tabelle in der Datenbank wird im Contract eine eigene Klasse angelegt, diese beinhaltet den Tabellennamen und die Namen aller ihrer Attribute als strings.

```
1  public static final class UebungenEntry implements
    BaseColumns {
2      public static final String TABLE_NAME = "uebungen";
3      public static final String COLUMN_ROWID = "_id";
4      public static final String COLUMN_NAME = "name";
5      public static final String COLUMN_MUSKELGRUPPE = "
    muskelgruppe";
6      public static final String COLUMN_BESCHREIBUNG = "
    beschreibung";
7      public static final String COLUMN_ANLEITUNG = "anleitung"
    ;
8      public static final String COLUMN_SCHWIERIGKEIT = "
    schwierigkeit";
9      public static final String COLUMN_VIDEO = "video";
10     public static final String COLUMN_EQUIPMENT = "equipment"
    ;
11 }
```