

SQLite

MIRIO EGGMANN & JAN ZIMMERMANN

Inhalt

- ▶ Was ist SQLite?
- ▶ Berechtigungen
- ▶ Wichtigste SQLite Klassen
- ▶ Aufbau
- ▶ Abfrage
- ▶ Weitere Informationen

Was ist SQLite?

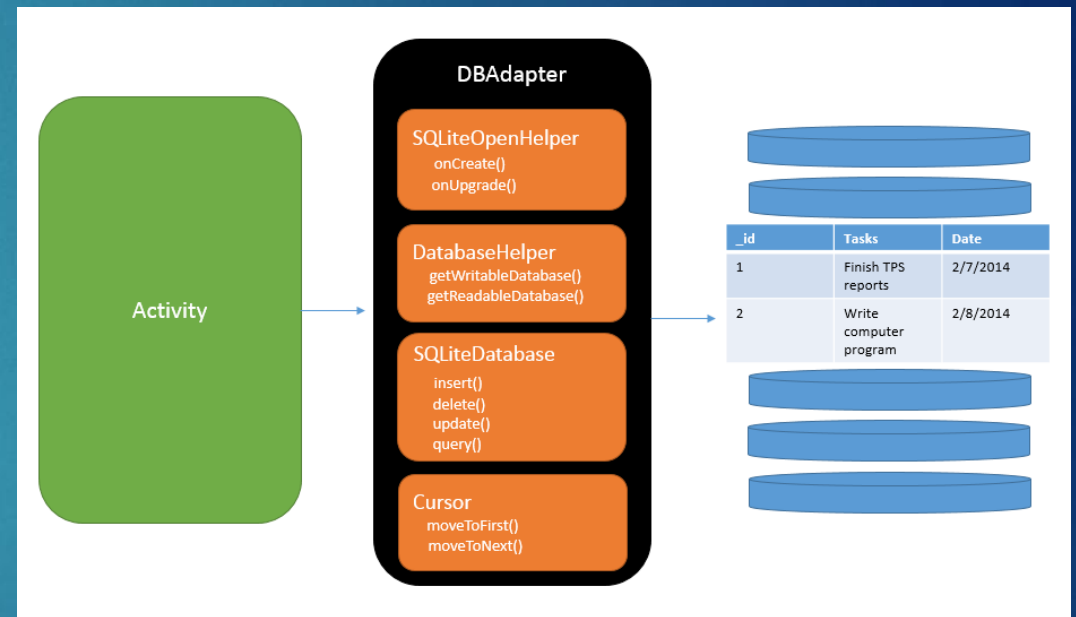
- ▶ Lokale relationale DB (kein Server)
- ▶ Eine DB Datei pro App (name.db)
- ▶ Optimal für mobile Geräte
 - ▶ Ressourcen sparend
 - ▶ Benötigt keinen Server
 - ▶ Portabel (nur eine Datei)
- ▶ Weniger Funktionen als z.B. MYSQL
 - ▶ RIGHT JOIN, FOR EACH, Weniger Datentypen
 - ▶ Die wichtigsten sind jedoch vorhanden

Berechtigungen

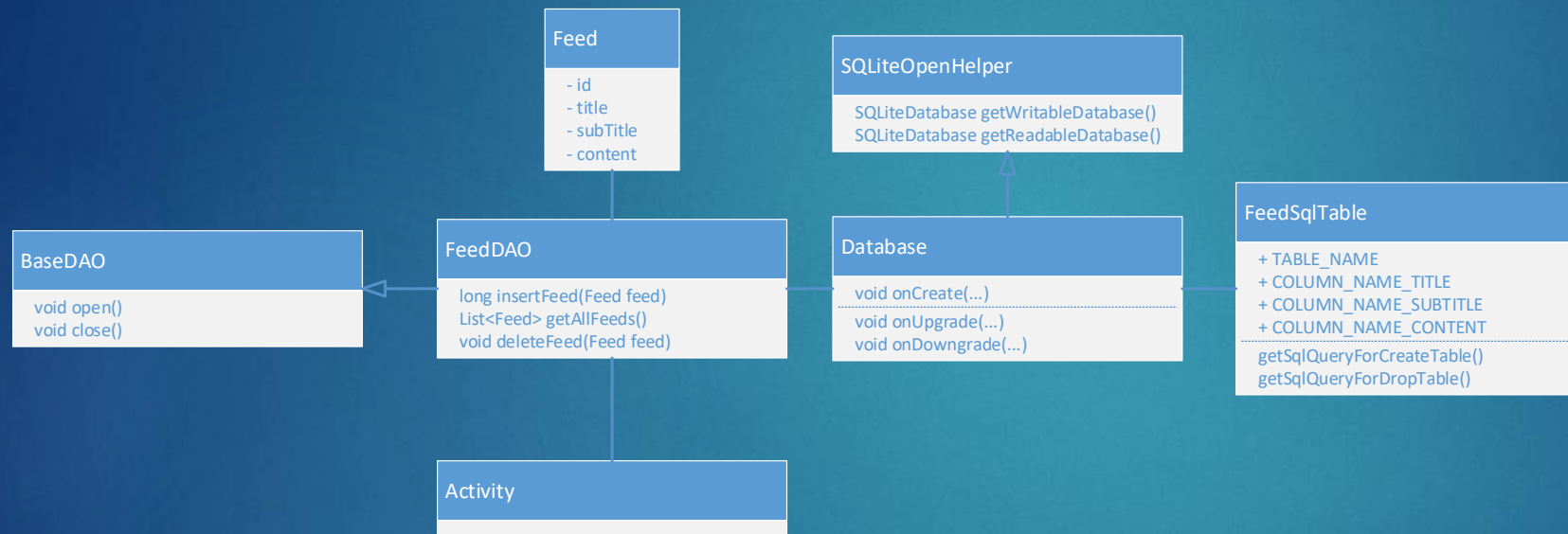
- ▶ Keine Benutzer und spezielle Berechtigungen
 - ▶ Lese- und Schreibberechtigungen, wie bei der Datei
- ▶ Auf die erstellte DB hat nur die dazugehörige Applikation Zugriff
 - ▶ DB Speicherort: /data/data/package.name/databases/<dbname>

Wichtigste SQLite Klassen

- ▶ SQLiteOpenHelper
 - ▶ Verwalten der DB (Create/Upgrade)
- ▶ SQLiteDatabase
 - ▶ Enthält Methoden zur Verwaltung der DB
- ▶ Cursor
 - ▶ Navigiert durch die Query Ergebnismenge
- ▶ SQLiteQueryBuilder
 - ▶ Bietet Hilfe zum erstellen von Queries



Aufbau



- ▼ **daos**
 - BaseDAO
 - ColorDAO
 - GradeDAO
 - GradetypeDAO
 - InstitutionDAO
 - PersonDAO
 - PersonSubjectDAO
 - SubjectDAO
 - SubjectGradeDAO
- ▼ **objects**
 - Color
 - Grade
 - Gradetype
 - Institution
 - Person
 - PersonSubject
 - Subject
 - SubjectGrade
- ▼ **sqltables**
 - BaseSqlTable
 - ColorSqlTable
 - GradeSqlTable
 - GradetypeSqlTable
 - InstitutionSqlTable
 - PersonSqlTable
 - PersonSubjectSqlTable
 - SubjectGradeSqlTable
 - SubjectSqlTable
- ▼ **values**
 - ColorValues
 - GradetypeValues
 - InstitutionValues
 - Database

DAO NOCH Löschen

```
► public class CommentDAO {  
    private final Database dbManager;  
    private SQLiteDatabase db;  
  
    public CommentDAO(Context context) {  
        dbManager = new Database(context);  
    }  
  
    public void open()  
    {  
        db = dbManager.getWritableDatabase();  
    }  
  
    public void close()  
    {  
        db.close();  
    }  
  
    public long insertComment(Comment comment) {  
        long newRowId = db.insert(...)  
        return newRowId;  
    }  
  
    public List<Comment> getAllComments() {  
        Cursor cursor = db.query(...);  
        ...  
    }  
}
```

Abfrage

SQL	SQLiteDatabase.query
<pre>INSERT INTO lernende (name, vorname) VALUES (‘Zimmermann’, ‘Jan’)</pre>	<pre>ContentValues werte = new ConentValues(); werte.put("name", "Zimmermann"); werte.put("vorname", "Jan") db.insert("lernende", null, werte);</pre>

Das Gleiche auch noch mit insert, update und delete

Weitere Informationen

9

- ▶ nullColumnHack

- ▶ Es darf nicht eine komplett leere Zeile eingefügt werden

```
INSERT INTO foo;
```

```
INSERT INTO foo (somecol) VALUES (NULL);
```

- ▶ getReadable() & getWritable()

- ▶ Kann nur aus der Datenbank lesen
 - ▶ Kann aus der Datenbank lesen & Daten einfügen

- ▶ Nach jedem Statement muss DB geschlossen werden



ANDROID

Fragen? प्रश्न?
Questions? 질문?
Ερωτήσεις; سوالات?
Αγγλικά?
Sorular? ಸ್ವಲ್ಪ?