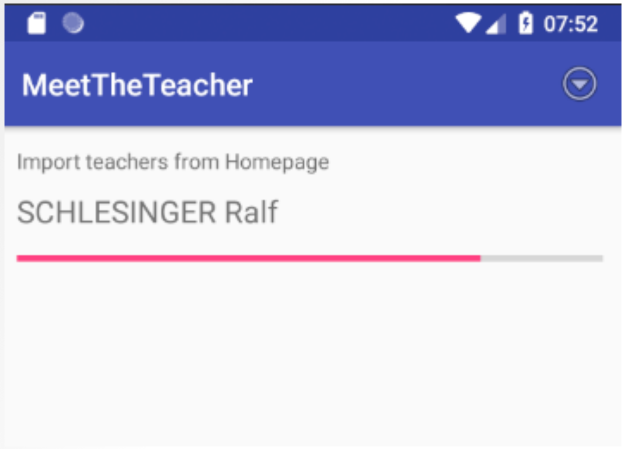
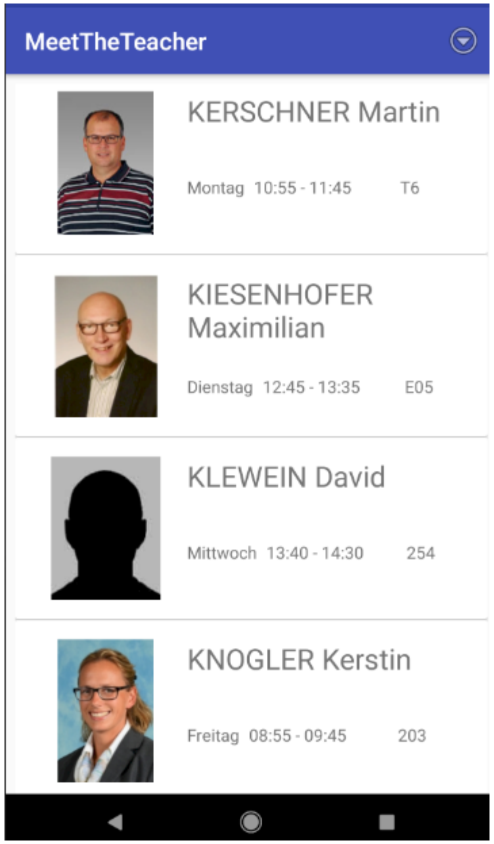
Persistenz in SQLITE-DB


07:52

MeetTheTeacher

Import teachers from Homepage

SCHLESINGER Ralf


MeetTheTeacher

KERSCHNER Martin
Montag 10:55 - 11:45 T6

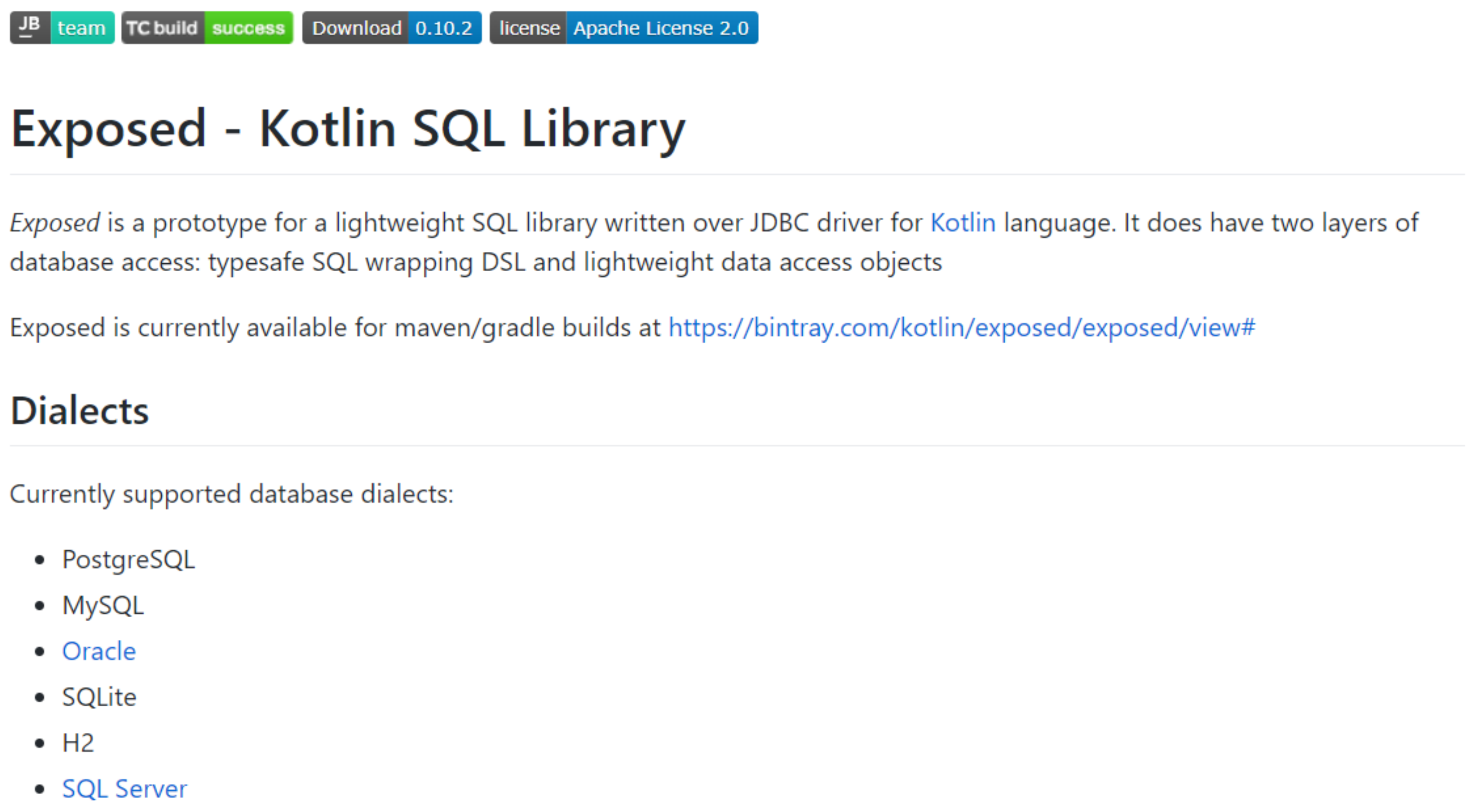
KIESENHOFER Maximilian
Dienstag 12:45 - 13:35 E05

KLEWEIN David
Mittwoch 13:40 - 14:30 254

KNOGLER Kerstin
Freitag 08:55 - 09:45 203

HTL LEONDING

OR-Mapper derzeit in Entwicklung



Exposed - Kotlin SQL Library

Exposed is a prototype for a lightweight SQL library written over JDBC driver for Kotlin language. It does have two layers of database access: typesafe SQL wrapping DSL and lightweight data access objects

Exposed is currently available for maven/gradle builds at https://bintray.com/kotlin/exposed/exposed/view#

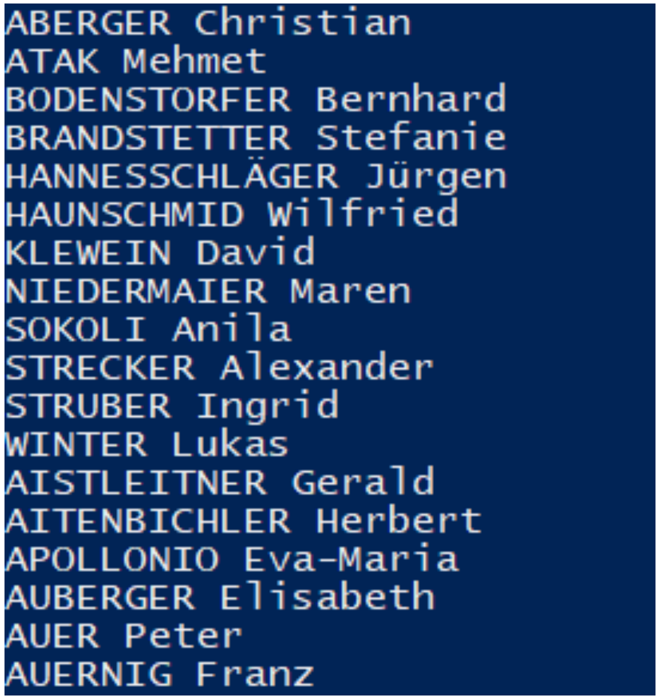
Dialects

Currently supported database dialects:

- PostgreSQL
- MySQL
- Oracle
- SQLite
- H2
- SQL Server

AddTeachers in DB implementieren

- Geladene Lehrerdaten in Datenbank speichern
- Nur eine Tabelle
  - Textspalten und eine BLOB-Spalte

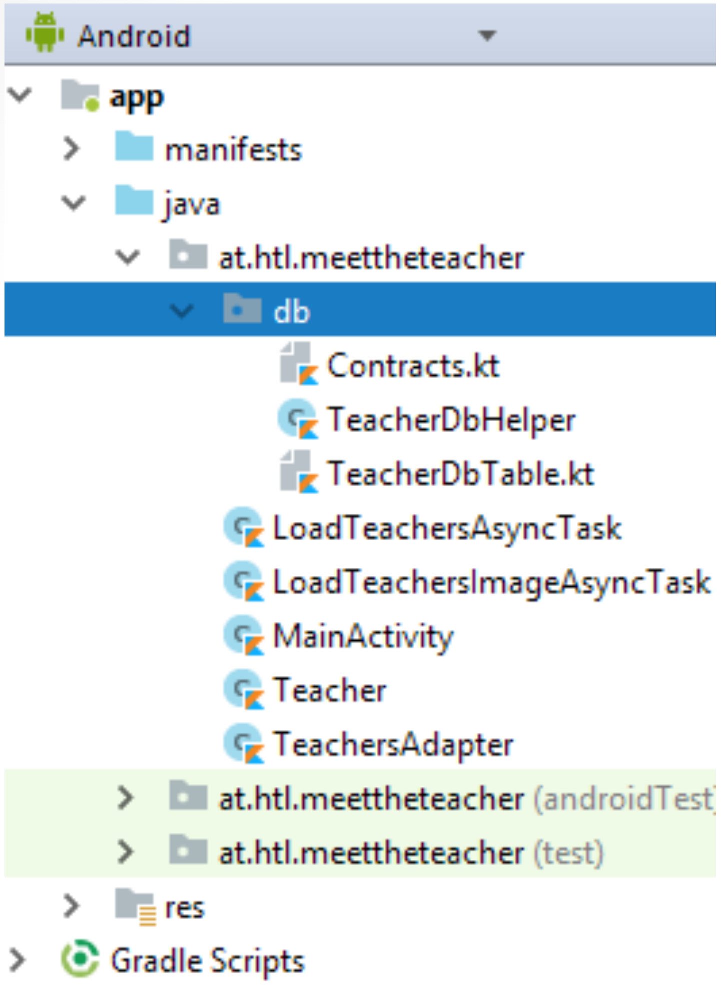


```
ABERGER Christian
ATAK Mehmet
BODENSTORFER Bernhard
BRANDSTETTER Stefanie
HANNESSCHLÄGER Jürgen
HAUNSCHMID Wilfried
KLEWEIN David
NIEDERMAIER Maren
SOKOLI Anila
STRECKER Alexander
STRUBER Ingrid
WINTER Lukas
AISTLEITNER Gerald
AITENBICHLER Herbert
APOLLONIO Eva-Maria
AUBERGER Elisabeth
AUER Peter
AUERNIG Franz
```

Übung

HTL LEONDING

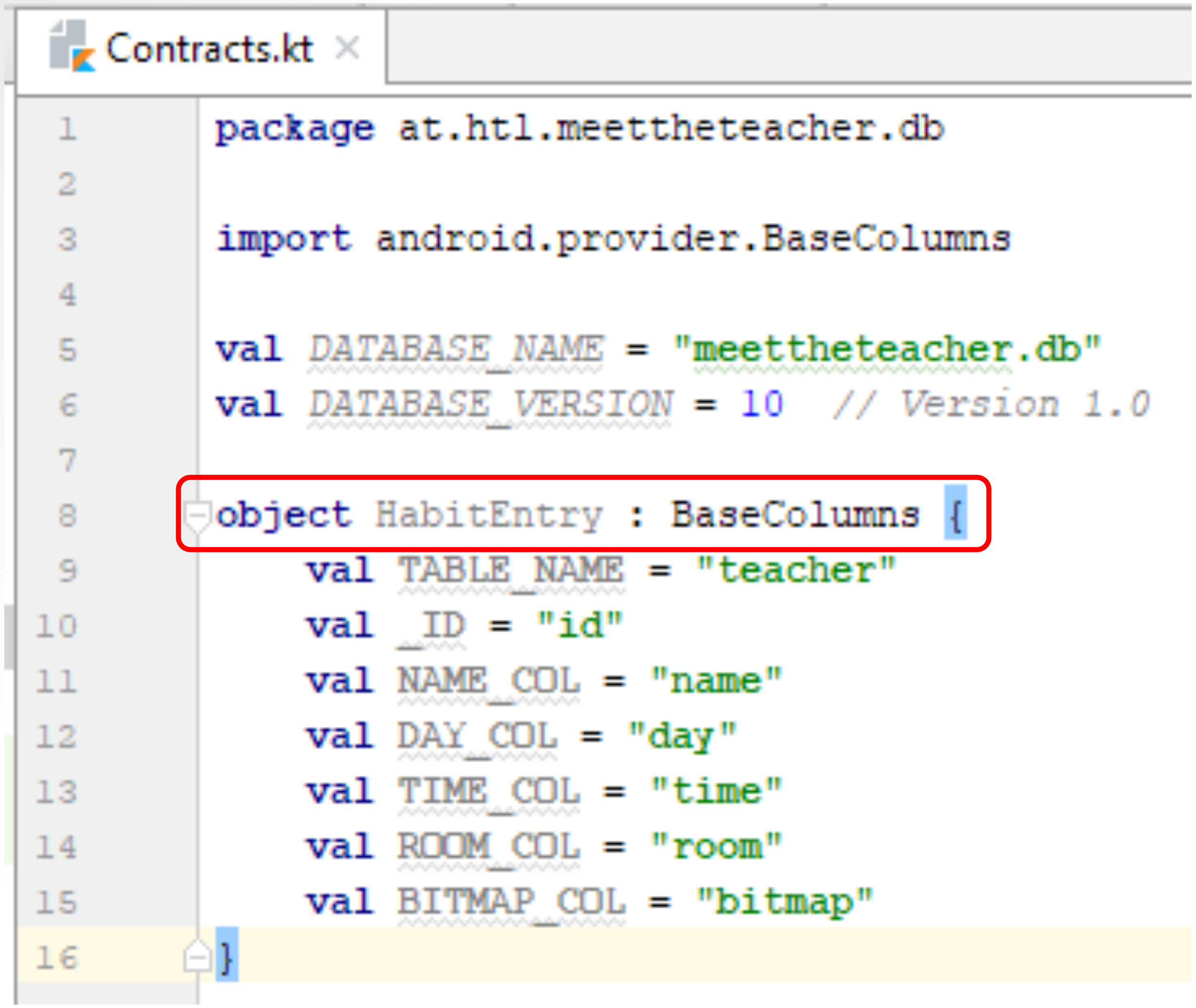
Eigenes Package für DB-Funktionalität



```
Android
app
  manifests
  java
    at.htl.meettheteacher
      db
        Contracts.kt
        TeacherDbHelper
        TeacherDbTable.kt
      LoadTeachersAsyncTask
      LoadTeachersImageAsyncTask
      MainActivity
      Teacher
      TeachersAdapter
    at.htl.meettheteacher (androidTest
    at.htl.meettheteacher (test)
  res
Gradle Scripts
```

HTL LEONDING

Contracts definieren

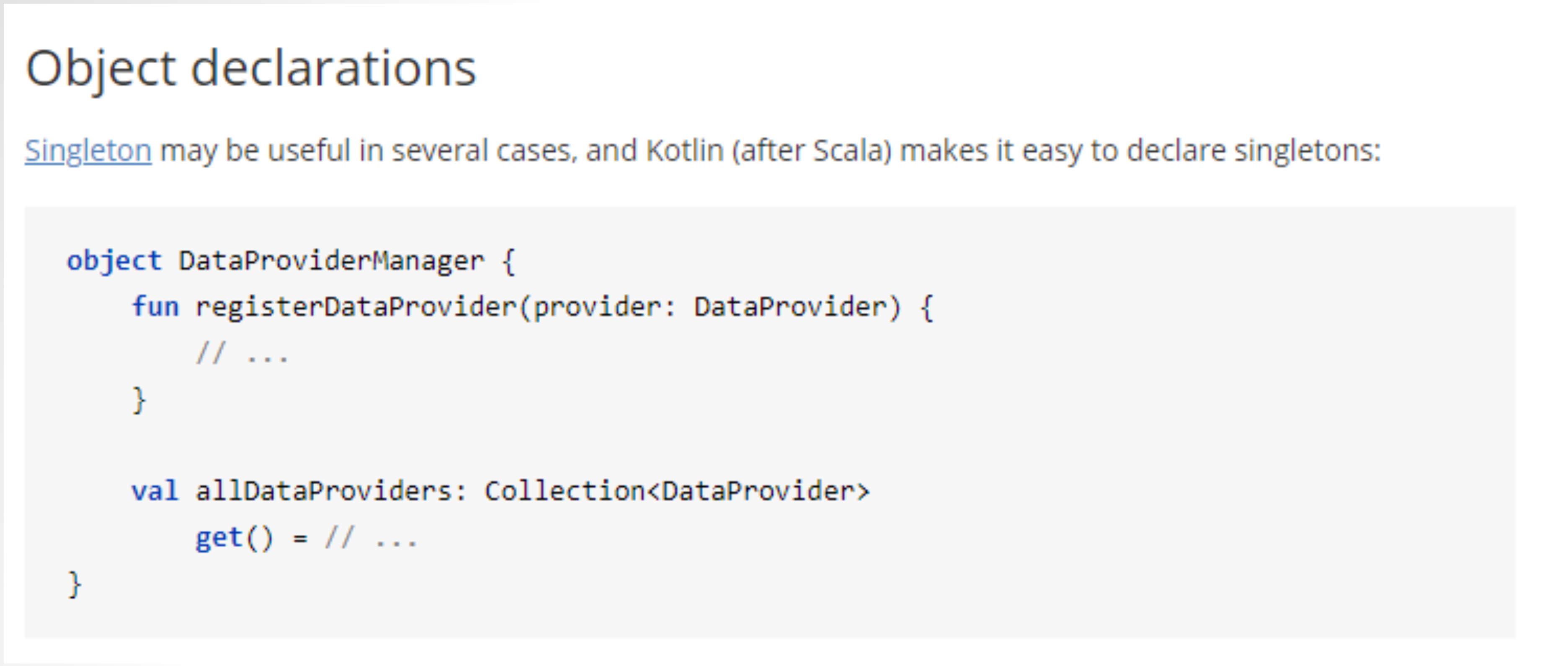


Contracts.kt

```kotlin
package at.htl.meettheteacher.db

import android.provider.BaseColumns

val DATABASE_NAME = "meettheteacher.db"
val DATABASE_VERSION = 10   // Version 1.0

object HabitEntry : BaseColumns {
    val TABLE_NAME = "teacher"
    val _ID = "id"
    val NAME_COL = "name"
    val DAY_COL = "day"
    val TIME_COL = "time"
    val ROOM_COL = "room"
    val BITMAP_COL = "bitmap"
}
```

HTL LEONDING

Kotlin - object-Declaration



Object declarations

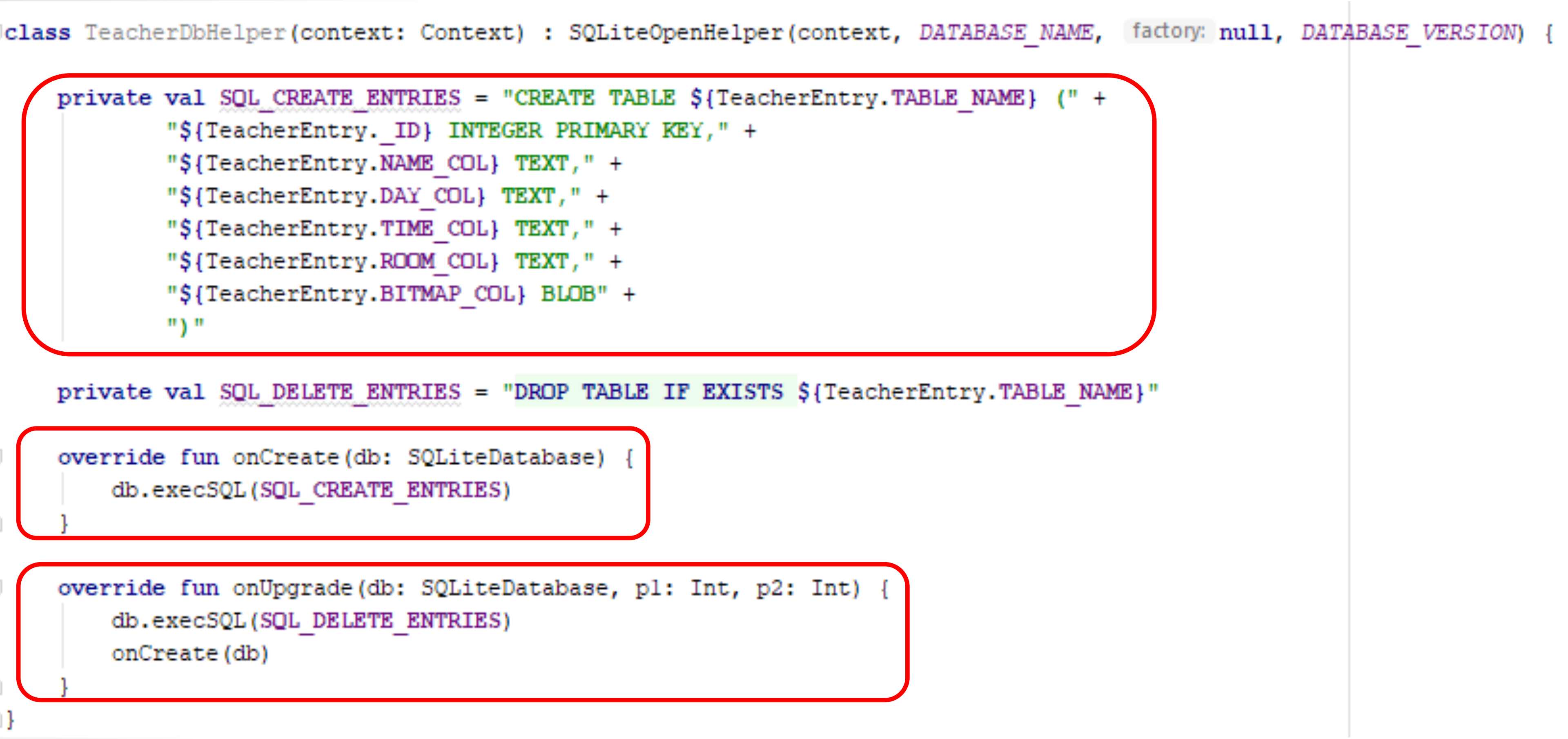
Singleton may be useful in several cases, and Kotlin (after Scala) makes it easy to declare singletons:

```kotlin
object DataProviderManager {
    fun registerDataProvider(provider: DataProvider) {
        // ...
    }

    val allDataProviders: Collection<DataProvider>
        get() = // ...
}
```

HTL LEONDING

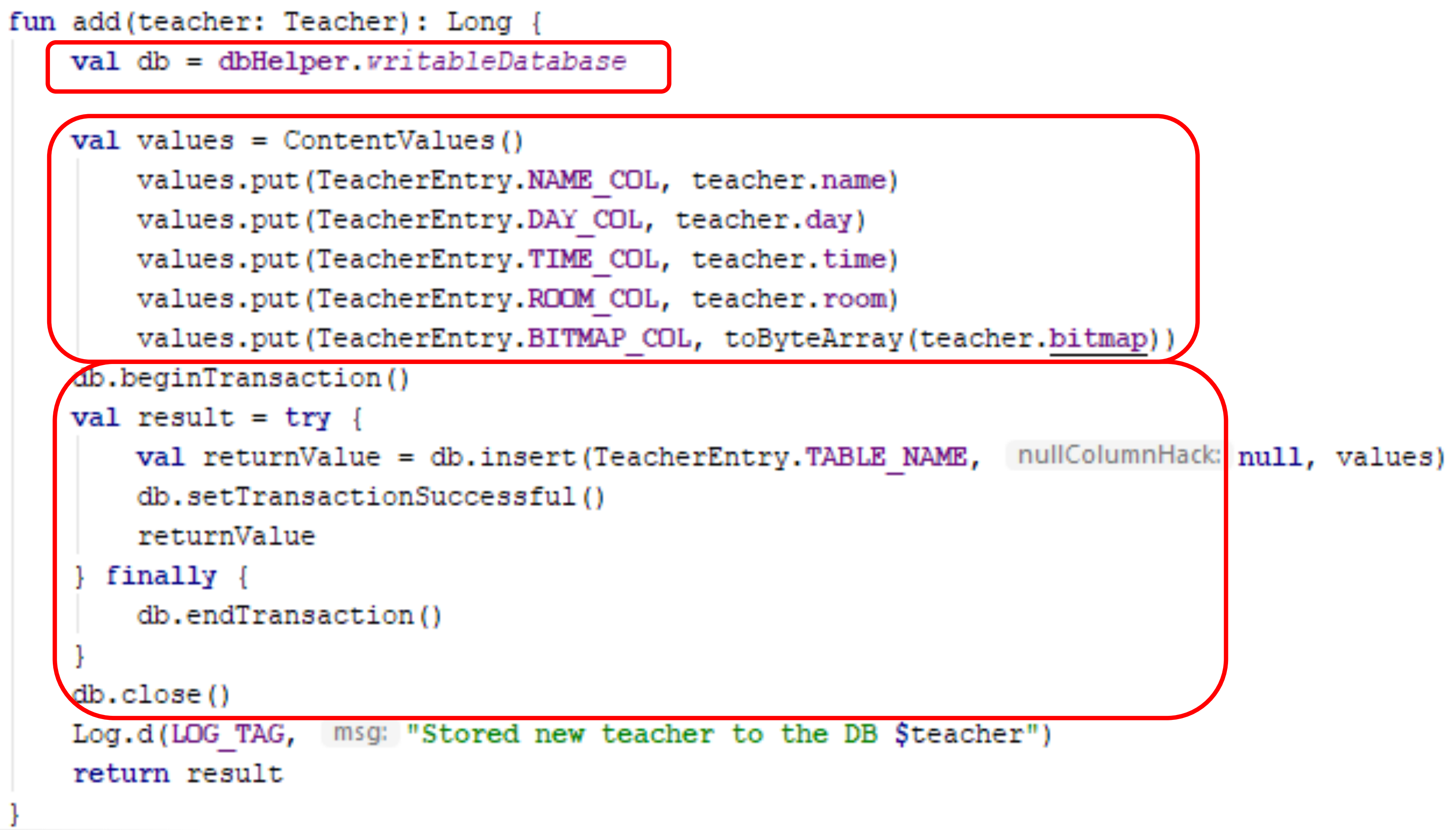
Datenbank verwalten → TeacherDbHelper

- SQL-Texte erzeugen
  - Stringinterpolation vereinfacht Handling
- Versionierung der Datenbank



```
class TeacherDbHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, factory: null, DATABASE_VERSION) {

    private val SQL_CREATE_ENTRIES = "CREATE TABLE ${TeacherEntry.TABLE_NAME} (" +
            "${TeacherEntry._ID} INTEGER PRIMARY KEY," +
            "${TeacherEntry.NAME_COL} TEXT," +
            "${TeacherEntry.DAY_COL} TEXT," +
            "${TeacherEntry.TIME_COL} TEXT," +
            "${TeacherEntry.ROOM_COL} TEXT," +
            "${TeacherEntry.BITMAP_COL} BLOB" +
            ")"

    private val SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS ${TeacherEntry.TABLE_NAME}"

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(SQL_CREATE_ENTRIES)
    }

    override fun onUpgrade(db: SQLiteDatabase, p1: Int, p2: Int) {
        db.execSQL(SQL_DELETE_ENTRIES)
        onCreate(db)
    }
}
```

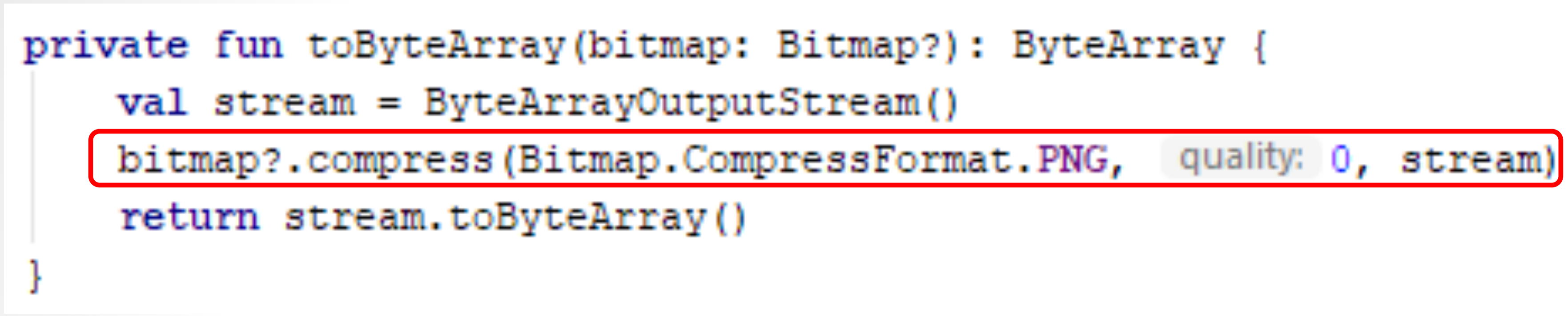
Add teacher – Java-Way



```
fun add(teacher: Teacher): Long {
    val db = dbHelper.writableDatabase

    val values = ContentValues()
        values.put(TeacherEntry.NAME_COL, teacher.name)
        values.put(TeacherEntry.DAY_COL, teacher.day)
        values.put(TeacherEntry.TIME_COL, teacher.time)
        values.put(TeacherEntry.ROOM_COL, teacher.room)
        values.put(TeacherEntry.BITMAP_COL, toByteArray(teacher.bitmap))
    db.beginTransaction()
    val result = try {
        val returnValue = db.insert(TeacherEntry.TABLE_NAME,  nullColumnHack: null, values)
        db.setTransactionSuccessful()
        returnValue
    } finally {
        db.endTransaction()
    }
    db.close()
    Log.d(LOG_TAG,  msg: "Stored new teacher to the DB $teacher")
    return result
}
```

Bitmap → ByteArray → BLOB → DB

- Bitmap ist nullable
  - Null-Safe-Operator
  - https://kotlinlang.org/docs/reference/null-safety.html

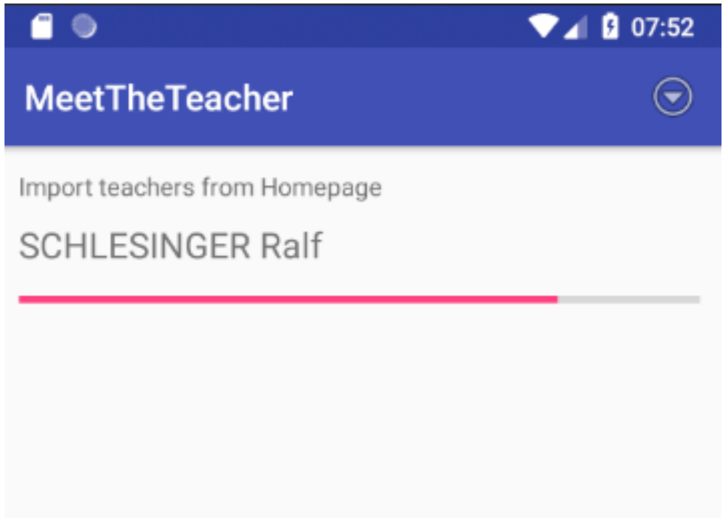


```kotlin
private fun toByteArray(bitmap: Bitmap?): ByteArray {
    val stream = ByteArrayOutputStream()
    bitmap?.compress(Bitmap.CompressFormat.PNG,  quality: 0, stream)
    return stream.toByteArray()
}
```

HTL LEONDING

Verwendung von addTeacher() in Mainactivity

- Auslösen über Menü
  - Siehe PayYours
  - Brauchbares Icon suchen
- View zwischen RecyclerView und Import-View wechseln
  - Visibility
  - Fragments
- Aktueller Lehrer im Download
  - Text oder CardView wiederverwenden
- Progressbar dokumentiert Fortschritt
  - Nur für Bilder
  - Text ist innerhalb einer Sekunde geladen



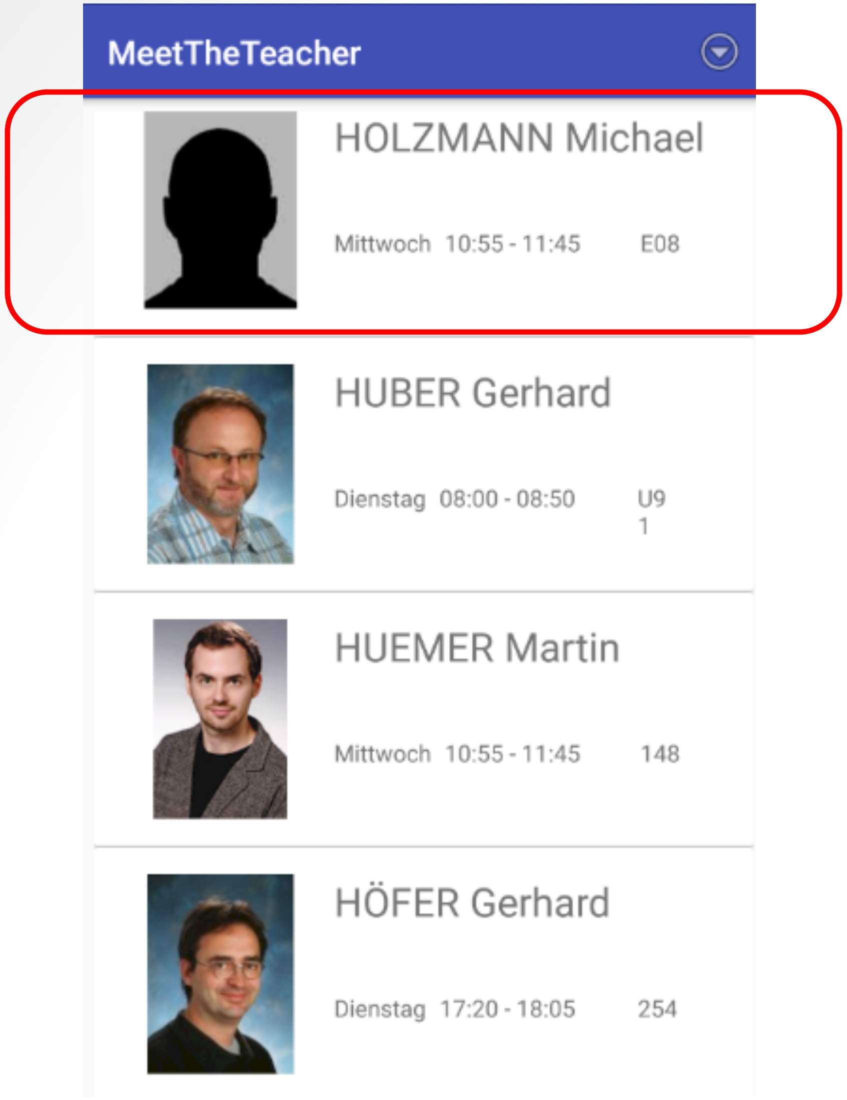
Import teachers from Homepage

SCHLESINGER Ralf

Übung

HTL LEONDING

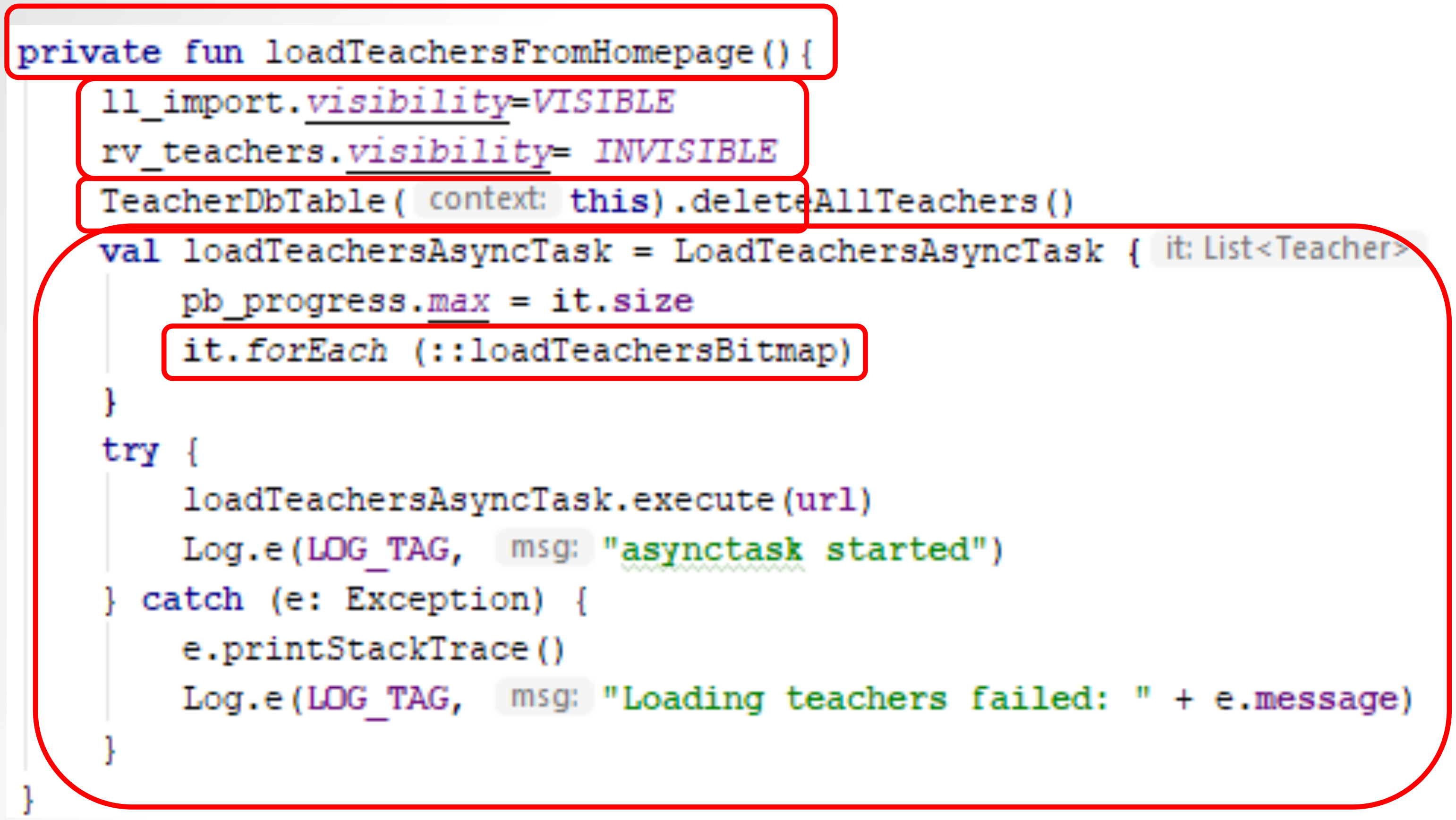
Bild trotz ID nicht verfügbar → Anonym



MeetTheTeacher

HOLZMANN Michael	Mittwoch	10:55 - 11:45	E08
HUBER Gerhard	Dienstag	08:00 - 08:50	U9 1
HUEMER Martin	Mittwoch	10:55 - 11:45	148
HÖFER Gerhard	Dienstag	17:20 - 18:05	254

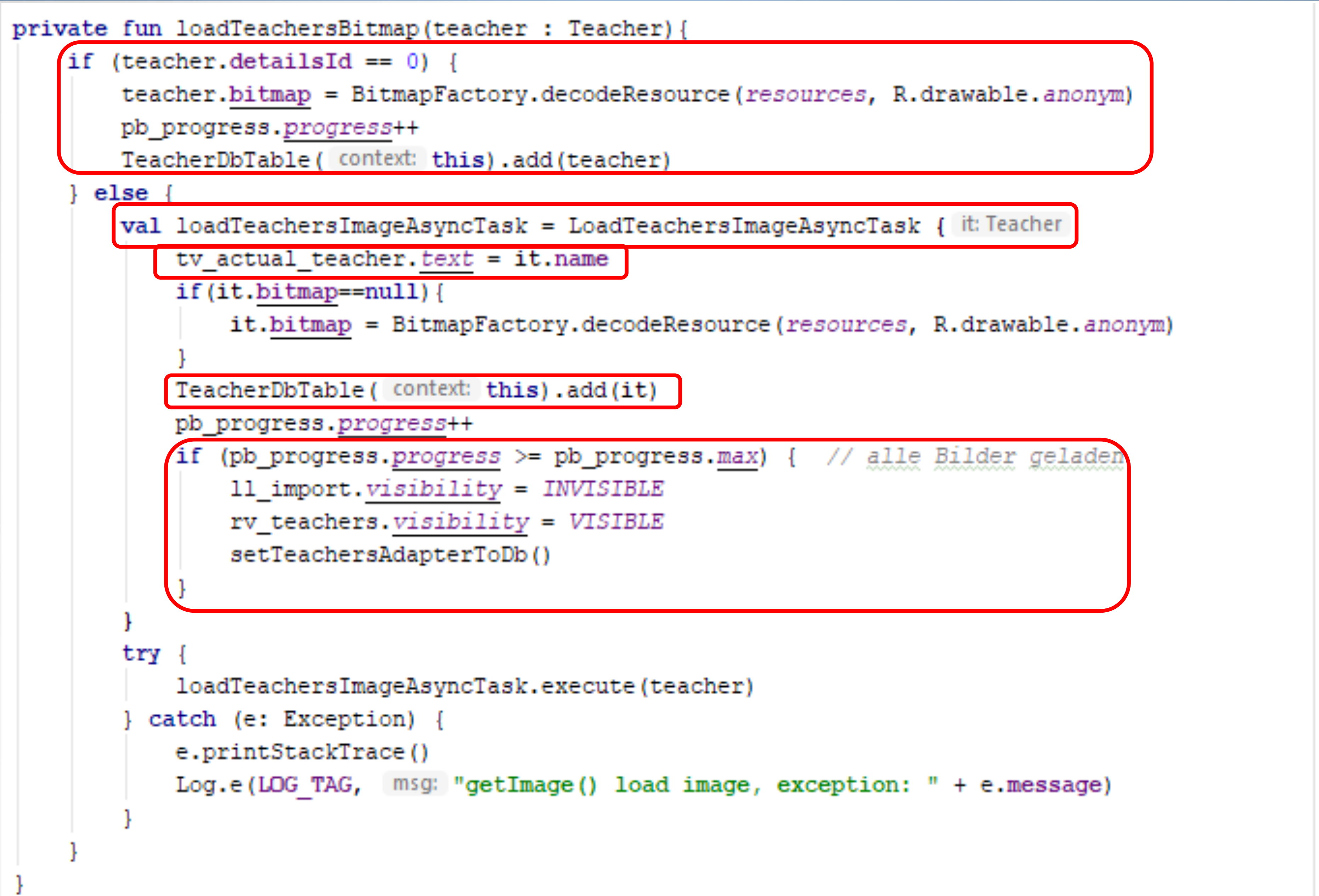
Menüpunkt ausgewählt



```
private fun loadTeachersFromHomepage(){
    ll_import.visibility=VISIBLE
    rv_teachers.visibility= INVISIBLE
    TeacherDbTable( context: this).deleteAllTeachers()
    val loadTeachersAsyncTask = LoadTeachersAsyncTask { it: List<Teacher>
        pb_progress.max = it.size
        it.forEach (::loadTeachersBitmap)
    }
    try {
        loadTeachersAsyncTask.execute(url)
        Log.e(LOG_TAG, msg: "asynctask started")
    } catch (e: Exception) {
        e.printStackTrace()
        Log.e(LOG_TAG, msg: "Loading teachers failed: " + e.message)
    }
}
```

HTL LEONDING

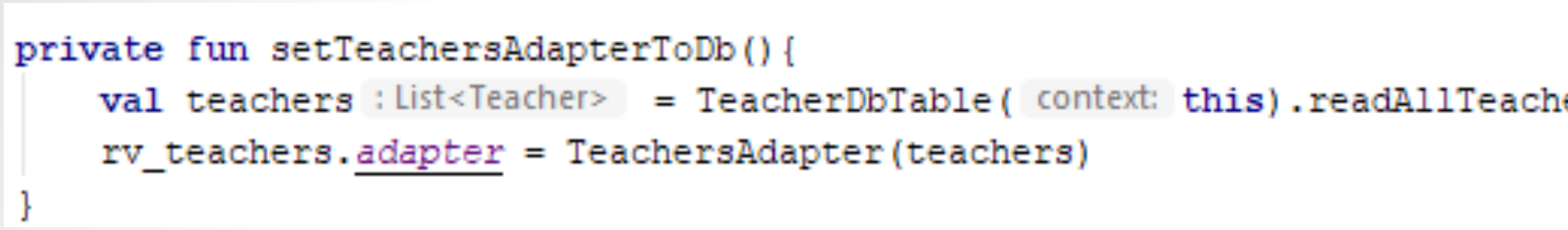
Bild des Lehrers laden



```
private fun loadTeachersBitmap(teacher : Teacher){
    if (teacher.detailsId == 0) {
        teacher.bitmap = BitmapFactory.decodeResource(resources, R.drawable.anonym)
        pb_progress.progress++
        TeacherDbTable( context: this).add(teacher)
    } else {
        val loadTeachersImageAsyncTask = LoadTeachersImageAsyncTask { it: Teacher
            tv_actual_teacher.text = it.name
            if(it.bitmap==null){
                it.bitmap = BitmapFactory.decodeResource(resources, R.drawable.anonym)
            }
            TeacherDbTable( context: this).add(it)
            pb_progress.progress++
            if (pb_progress.progress >= pb_progress.max) {  // alle Bilder geladen
                ll_import.visibility = INVISIBLE
                rv_teachers.visibility = VISIBLE
                setTeachersAdapterToDb()
            }
        }
        try {
            loadTeachersImageAsyncTask.execute(teacher)
        } catch (e: Exception) {
            e.printStackTrace()
            Log.e(LOG_TAG,  msg: "getImage() load image, exception: " + e.message)
        }
    }
}
```

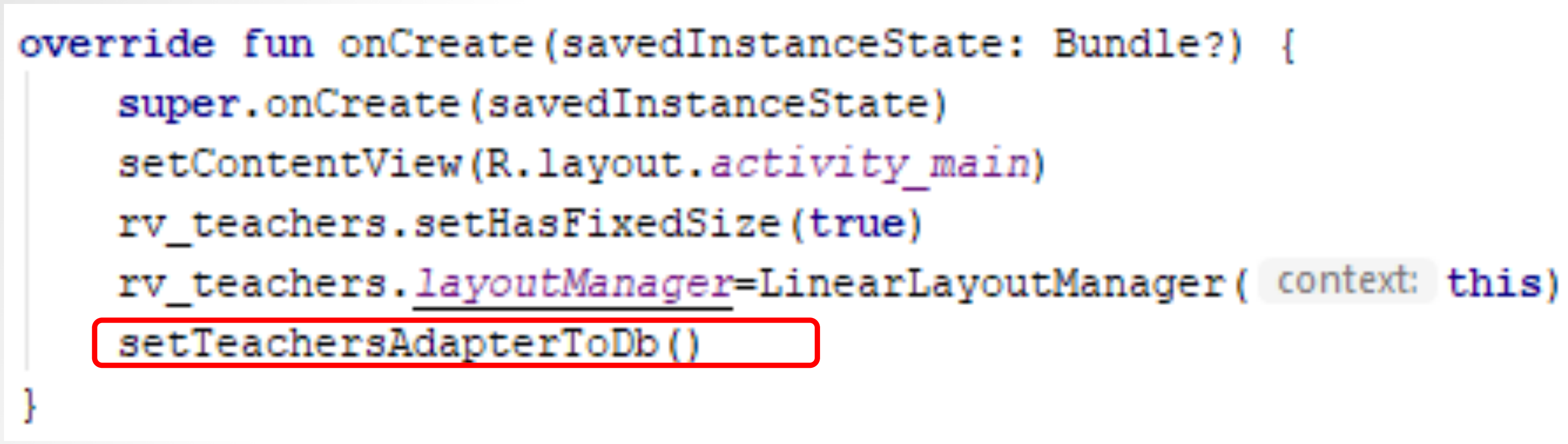
RecyclerView-Adapter versorgen

- TeacherDbTable um readAllTeachers() erweitern



```
private fun setTeachersAdapterToDb(){
    val teachers : List<Teacher>  = TeacherDbTable( context: this).readAllTeache
    rv_teachers.adapter = TeachersAdapter(teachers)
}
```

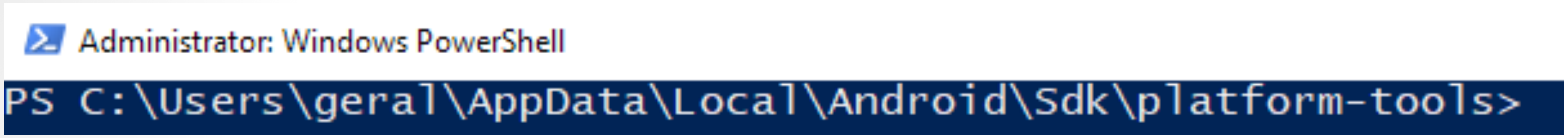
Bei App-Start Daten aus DB laden



```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    rv_teachers.setHasFixedSize(true)
    rv_teachers.layoutManager=LinearLayoutManager( context: this)
    setTeachersAdapterToDb()
}
```

Kontrolle der Daten in SQLITE-DB

- Datenbankdatei aus Anwendungsverzeichnis auf PC kopieren
- Mit Commandline-Tools inspizieren
- Wechsel in Tools-Verzeichnis von Android

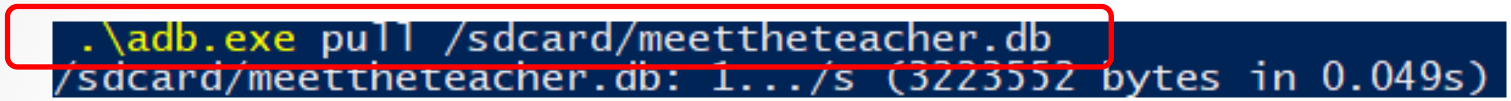

Administrator: Windows PowerShell

```
PS C:\Users\geral\AppData\Local\Android\Sdk\platform-tools>
```

HTL LEONDING

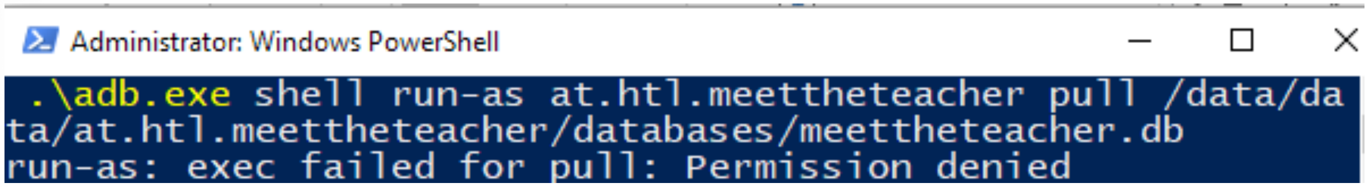
Datenbankdatei herunterladen

- Zugriff mit Credentials der App
- Datenbank zuerst auf SD-Karte kopieren
  - .\adb.exe shell run-as at.htl.meettheteacher cp /data/data/at.htl.meettheteacher/databases/meettheteacher.db /sdcard/
- Daten von SD-Karte auf PC übertragen
  - .\adb.exe pull /sdcard/meettheteacher.db



```
.\adb.exe pull /sdcard/meettheteacher.db
/sdcard/meettheteacher.db: 1.../s (3223552 bytes in 0.049s)
```

  - Direktes pull aus Appverzeichnis ist nicht erlaubt



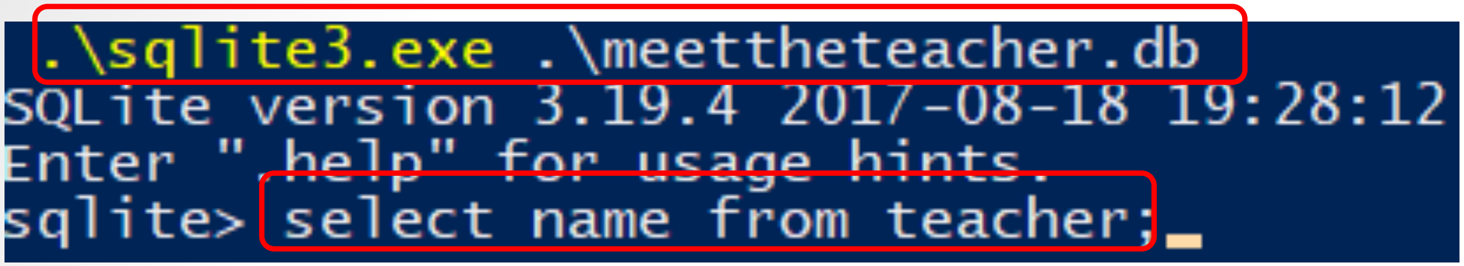
Administrator: Windows PowerShell

```
.\adb.exe shell run-as at.htl.meettheteacher pull /data/da
ta/at.htl.meettheteacher/databases/meettheteacher.db
run-as: exec failed for pull: Permission denied
```

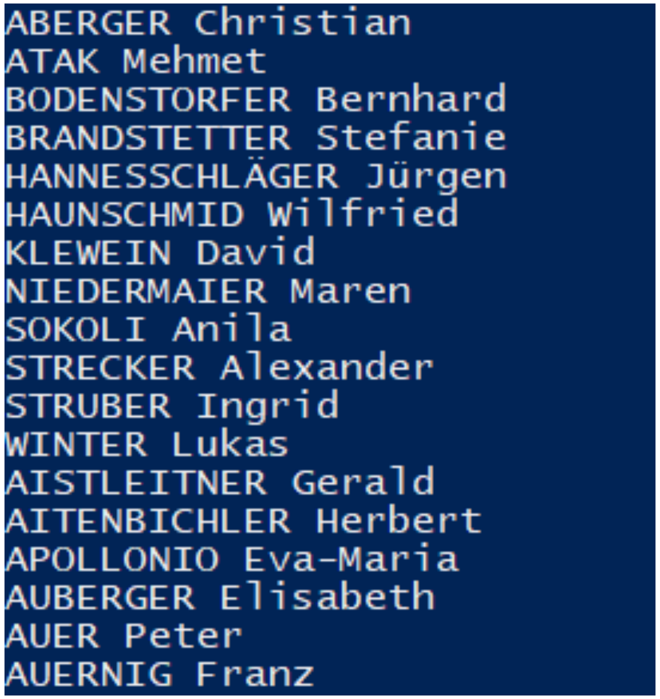
HTL LEONDING

Datenbankdatei analysieren

- SQLITE-Commandline



```
.\sqlite3.exe .\meettheteacher.db
SQLite version 3.19.4 2017-08-18 19:28:12
Enter ".help" for usage hints.
sqlite> select name from teacher;
```

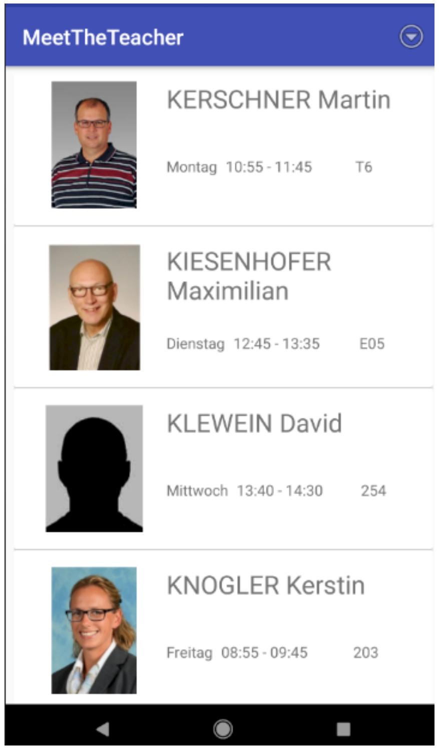


```
ABERGER Christian
ATAK Mehmet
BODENSTORFER Bernhard
BRANDSTETTER Stefanie
HANNESSCHLÄGER Jürgen
HAUNSCHMID Wilfried
KLEWEIN David
NIEDERMAIER Maren
SOKOLI Anila
STRECKER Alexander
STRUBER Ingrid
WINTER Lukas
AISTLEITNER Gerald
AITENBICHLER Herbert
APOLLONIO Eva-Maria
AUBERGER Elisabeth
AUER Peter
AUERNIG Franz
```

HTL LEONDING

Aufgaben

- TeacherDbTable erweitern
  - readAllTeachers() liest alle Lehrer sortiert nach Namen aus der Datenbank
    - Hinweis: db.query() liefert Cursor zurück
  - deleteAllTeacher() löscht alle Lehrer aus der Datenbank
    - Aufruf vor Start des Imports
- Soweit möglich, Kotlin-Features einsetzen
  - Extension-Functions, spezielle Functions (with, …)



KERSCHNER Martin
Montag 10:55 - 11:45 T6

KIESENHOFER Maximilian
Dienstag 12:45 - 13:35 E05

KLEWEIN David
Mittwoch 13:40 - 14:30 254

KNOGLER Kerstin
Freitag 08:55 - 09:45 203

Übung

HTL LEONDING