



JÖNKÖPING UNIVERSITY

School of Engineering

ANDROID SQLITE DATABASE

Peter Larsson-Green

Jönköping University

Spring 2020

WHY STORE DATA IN A DATABASE?

```
public class Data{  
    public static List<String> myWords = new ArrayList<>();  
    static{  
        myWords.add("One");  
        myWords.add("Two");  
        myWords.add("Three");  
    }  
}
```

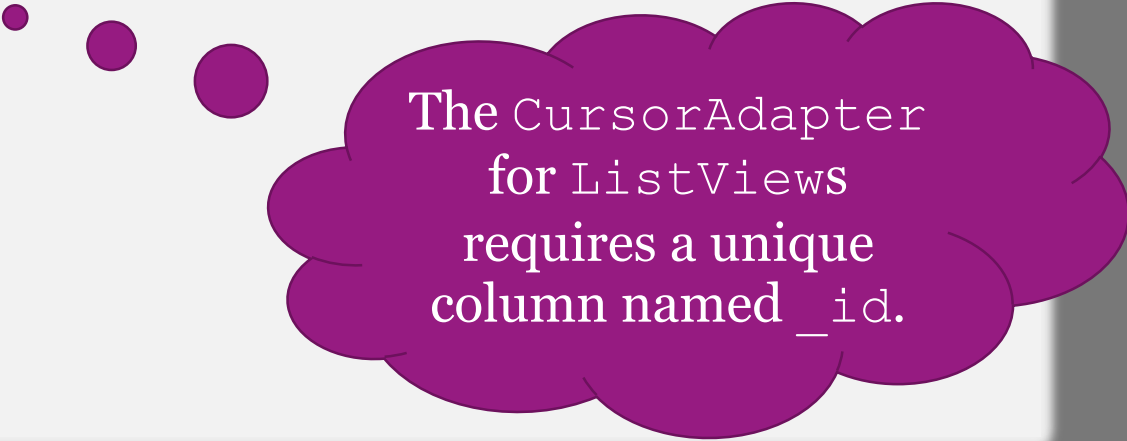
- OK for static data.
- Not OK for dynamic data.
 - Application stops → All data is lost 😞

DATABASE HELPER

```
public class DatabaseHelper extends SQLiteOpenHelper{  
    public DatabaseHelper(Context aContext) {  
        super(aContext, "my-database.db", null, 1);  
    }  
    public void onCreate(SQLiteDatabase db) {  
    }  
    public void onUpgrade(SQLiteDatabase db, int oldVer, int newVer) {  
    }  
    public void onDowngrade(SQLiteDatabase db, int oldVer, int newVer) {  
    }  
}
```

DATABASE CREATE

```
public class DatabaseHelper extends SQLiteOpenHelper{  
    public void onCreate(SQLiteDatabase db){  
        db.execSQL(  
            "CREATE TABLE humans (" +  
            "_id INTEGER PRIMARY KEY AUTOINCREMENT," +  
            "name TEXT" +  
            ") "  
        );  
    }  
}
```



The CursorAdapter
for ListViews
requires a unique
column named `_id`.

DATABASE INSERT

```
DatabaseHelper dbHelper = new DatabaseHelper(aContext);
SQLiteDatabase db = dbHelper.getWritableDatabase();
db.execSQL(
    "INSERT INTO humans (name) VALUES (?)",
    new String[]{ "Allen" }
);
```

```
ContentValues values = new ContentValues();
values.put("name", "Allen");
long id = db.insert("humans", null, values);
```

DATABASE READ

```
DatabaseHelper databaseHelper = new DatabaseHelper(aContext);
SQLiteDatabase db = databaseHelper.getReadableDatabase();
Cursor cursor = db.rawQuery(
    "SELECT _id, name FROM humans WHERE _id < ?",
    new String[]{ "10" }
);
while (cursor.moveToNext()) {
    long _id = cursor.getLong(0);
    String name = cursor.getString(1);
}
cursor.close()
```

DATABASE UPDATE

```
DatabaseHelper databaseHelper = new DatabaseHelper(aContext);  
SQLiteDatabase db = databaseHelper.getWritableDatabase();  
db.execSQL(  
    "UPDATE humans SET name = ? WHERE _id = ? LIMIT 1",  
    new String[]{ "Newell", "1" }  
);
```

```
ContentValues values = new ContentValues();  
values.put("name", "Allen");  
db.update("humans", values, "_id = ?", new String[]{ "1" });
```


DATABASE DELETE

```
DatabaseHelper databaseHelper = new DatabaseHelper(aContext);  
SQLiteDatabase db = databaseHelper.getWritableDatabase();  
db.execSQL(  
    "DELETE FROM humans WHERE _id = ? LIMIT 1",  
    new String[]{ "1" }  
);
```

```
db.delete("humans", "_id = ?", new String[]{ "1" });
```

CLOSING A DATABASE

Databases should be closed when not used anymore.

```
DatabaseHelper dbHelper = new DatabaseHelper(aContext);  
SQLiteDatabase db = dbHelper.getWritableDatabase();  
// Do your thing...  
db.close();
```

COMMON DB PATTERN

Create a single global database instance and never close it.

- Use the singleton pattern.

COMMON DB PATTERN

```
public class DatabaseHelper extends SQLiteOpenHelper{  
    private static SQLiteDatabase db = null;  
    public static SQLiteDatabase getInstance(Context aContext){  
        if(db == null){  
            db = new DatabaseHelper(aContext).getWritableDatabase();  
        }  
        return db;  
    }  
    private DatabaseHelper(Context aContext){  
        super(aContext, "my-database.db", null, 1);  
    }  
    // Implement onCreate(), onUpgrade() and onDowngrade().  
}
```

Use
aContext.getApplicationContext()
to avoid memory leaks.