



Как я возненавидел ORM

Сергей Боиштян, Тинькофф

Мы будем говорить

Про Orm

Про Sqlite

Про ненависть

Orm?

Object-Relational Mapping

Добро пожаловать в мир разработки

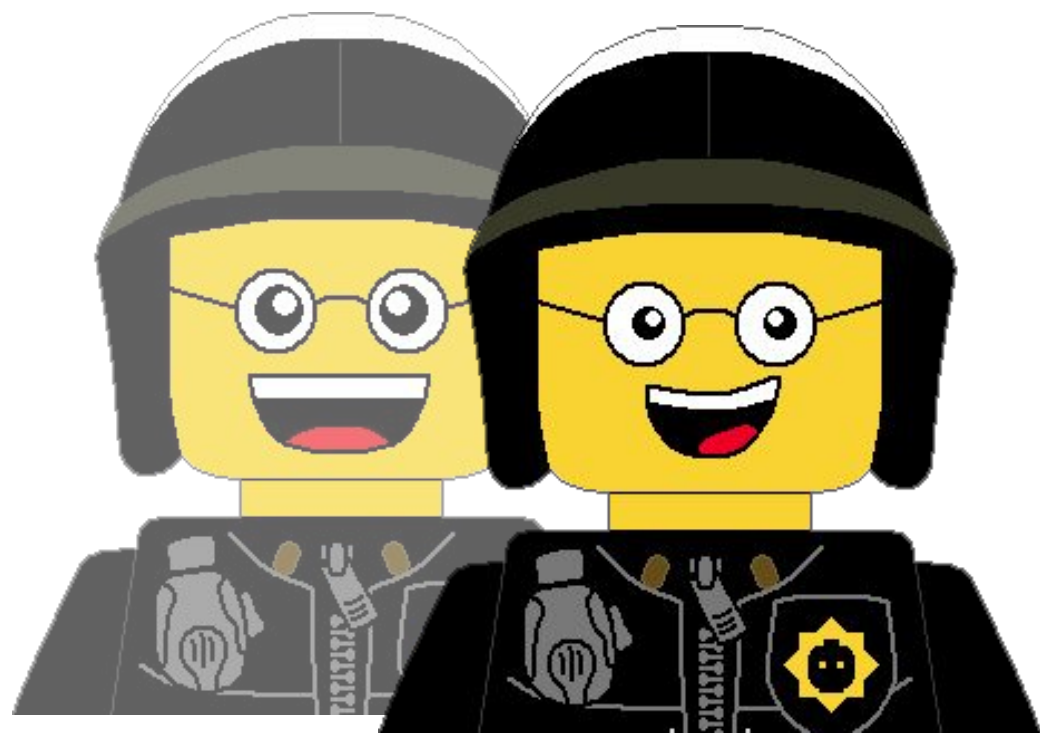


Создание db

```
private static final String TEXT_TYPE = " TEXT";  
private static final String COMMA_SEP = ",";  
private static final String SQL_DELETE_ENTRIES =  
    "DROP TABLE IF EXISTS " + FeedEntry.TABLE_NAME;  
private static final String SQL_CREATE_ENTRIES =  
    "CREATE TABLE " + FeedEntry.TABLE_NAME + " (" +  
    FeedEntry._ID + " INTEGER PRIMARY KEY," +  
    FeedEntry.COLUMN_NAME_ENTRY_ID + TEXT_TYPE + COMMA_SEP +  
    FeedEntry.COLUMN_NAME_TITLE + TEXT_TYPE + COMMA_SEP +  
    ... // Any other options for the CREATE command " );
```

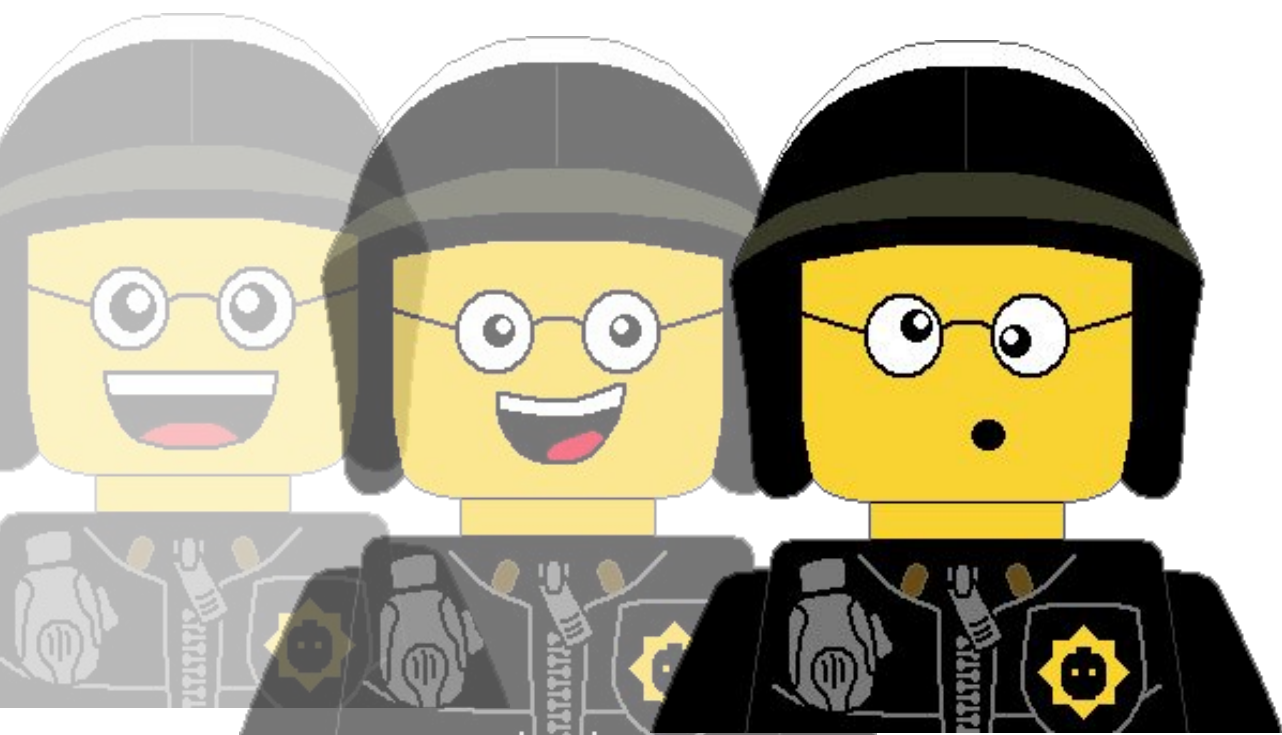
Создание db

```
public class FeedReaderDbHelper extends SQLiteOpenHelper {  
    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) {  
        db.execSQL(SQL_DELETE_ENTRIES);  
        onCreate(db);  
    }  
}
```



Запись в бд

```
SQLiteDatabase db = mDbHelper.getWritableDatabase();  
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);  
long newRowId = db.insert(FeedEntry.TABLE_NAME,  
FeedEntry.COLUMN_NAME_NULLABLE, values);
```

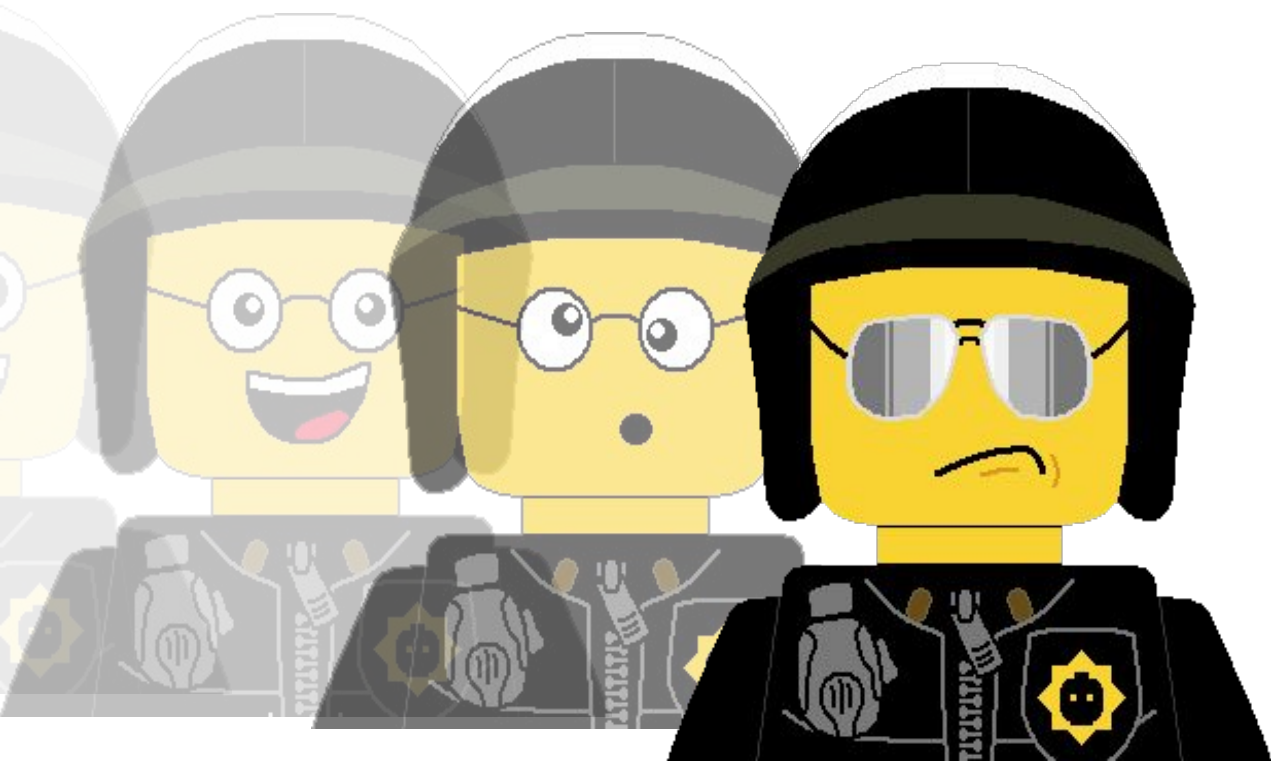



Удаление из бд

```
String selection = FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";
```

```
String[] selectionArgs = { String.valueOf(rowId) };
```

```
db.delete(table_name, selection, selectionArgs);
```



Чтение из бд

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

```
String[] projection = {
```

```
    FeedEntry._ID,
```

```
    FeedEntry.COLUMN_NAME_TITLE,
```

```
    FeedEntry.COLUMN_NAME_UPDATED,
```

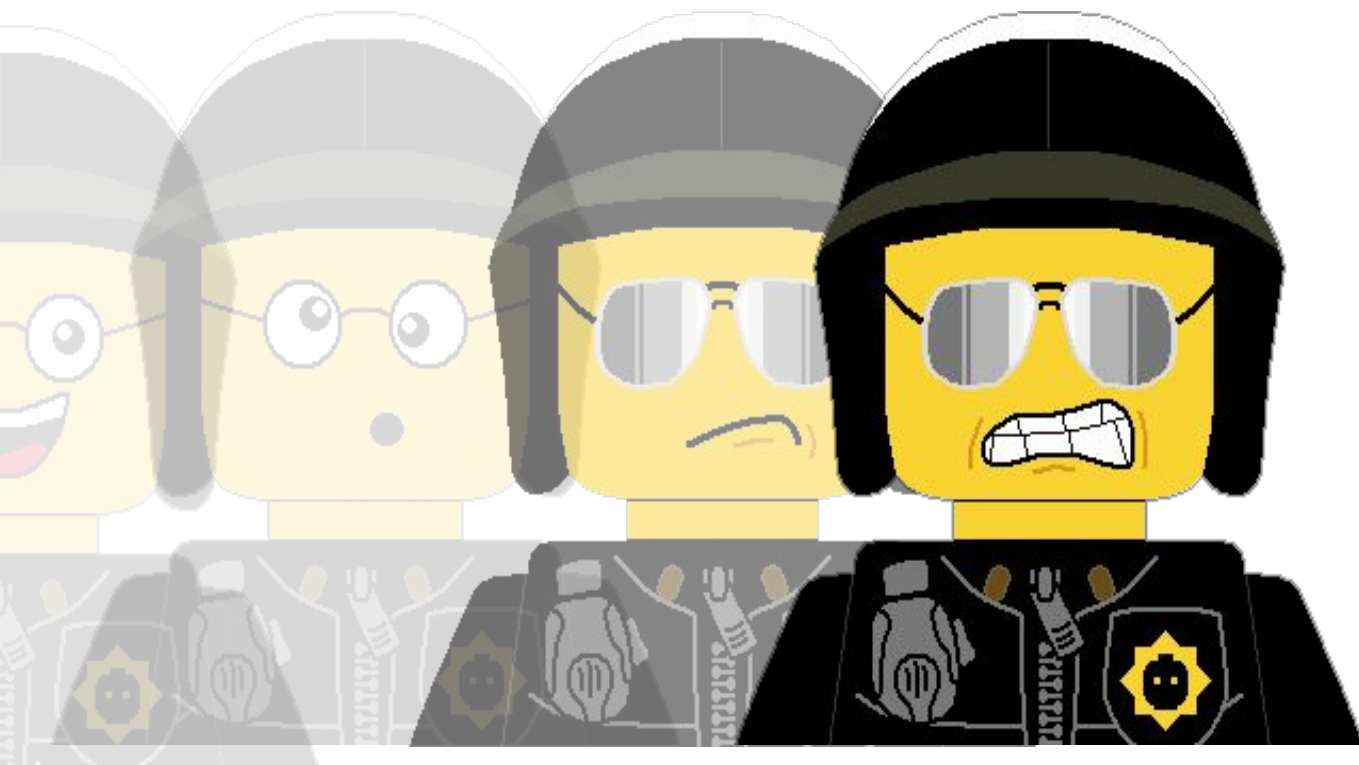
```
    ... };
```

```
String sortOrder = FeedEntry.COLUMN_NAME_UPDATED + " DESC";
```

Чтение из бд

```
Cursor c = db.query(  
    FeedEntry.TABLE_NAME,  
    projection,  
    selection,  
    selectionArgs,  
    null,  
    null,  
    sortOrder);
```

“Теперь ты Senior, сынок”



Orm?



Причины использования Orm

Несоответствие SQL и ООП парадигм

ООП путь

Ускорение разработки

Простота

Типобезопасность

Несоответствие парадигм

Гранулярность

Наследование

Идентичность

Ассоциативность

Навигация по данным

Гранулярность

OOP

```
class User {  
    String name;  
    Address address;  
}
```

SQL

```
TABLE USER (  
    name, city, country);
```

Наследование

OOP

```
class User {  
    String name;  
}  
  
class Student extends User{}  
class Teacher extends User{}
```

SQL

нет нативной поддержки

Идентичность

OOP

`object.equals(object)`

`object == object`

SQL

Primary key identity

Ассоциативность

OOP

```
class User {  
    Address adress;  
}  
  
class Address {  
    User user;  
}
```

SQL

```
TABLE USER (  
id, name, address_id (Fk) );  
  
TABLE ADDRESS (  
address_id, city, country);
```

ООП путь

Инкапсуляция от низкоуровневых API

Наши модели могут быть объектами, а не структурами

Ускорение разработки

Маппинг данных

Ленивая инициализация

Уменьшение кол-ва boilerplate

Простота

Не нужно больше писать сложные SQL запросы

Типобезопасность

Меньше строк(String) больше кода

Я и Орм



**BEST FRIENDS ALWAYS
STICK TOGETHER**

Дружба - это принимать друг друга

@DatabaseField, 29 параметров

Дружба - это принимать друг друга

@DatabaseTable

```
public class PaymentDocuemnt {
```

```
    @DatabaseField(generatedId = true)
```

```
    public long id;
```

```
    @DatabaseField(canBeNull = false,
```

```
                    dataType = DataType.SERIALIZABLE)
```

```
    private CodeValue codeValue;
```

```
}
```

HO...

Постановка задачи

Исследовать разрастание БД в проекте

Description

Click to add description

Воспроизводим

Что измерить?

Каким способом получить измеряемые данные?

Шаги?

Что измерить?

Размер файла бд



Каким способом получить измеряемые данные?

```
File f = context.getDatabasePath(dbName);  
long dbSize = f.length();
```

or

```
pragma page_size; * pragma page_count;
```

Pragma

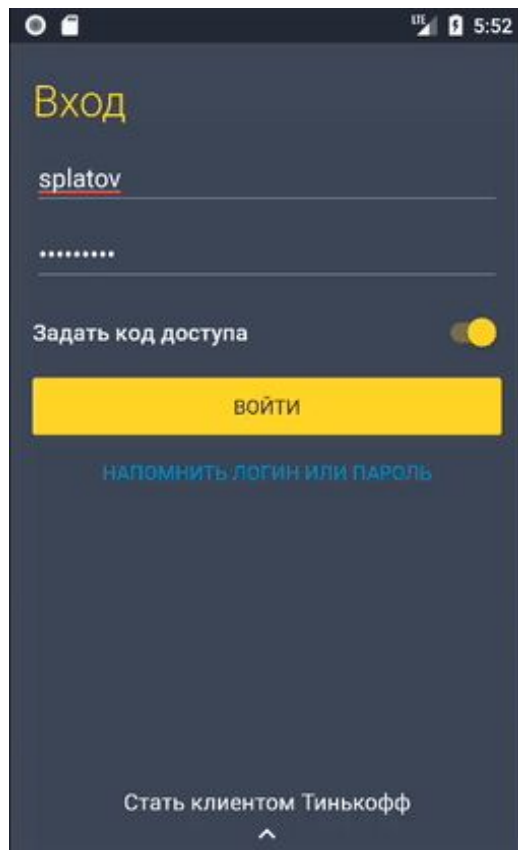
Statement специфичный для Sqlite

Модифицирование настроек Sqlite

Получение метаданных о бд

<https://sqlite.org/pragma.html>

Шаги? (Flow авторизации)



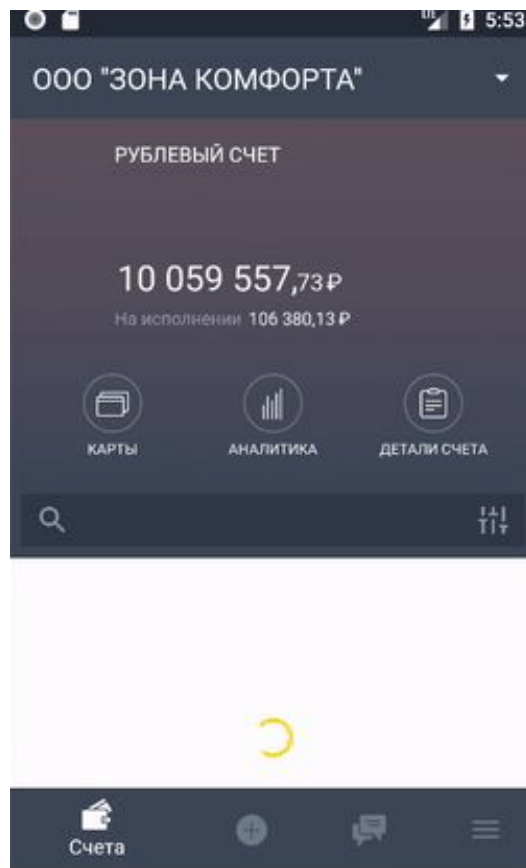
Шаги? Смена компании



Шаги? Платежи



Шаги? Выход до пин-кода



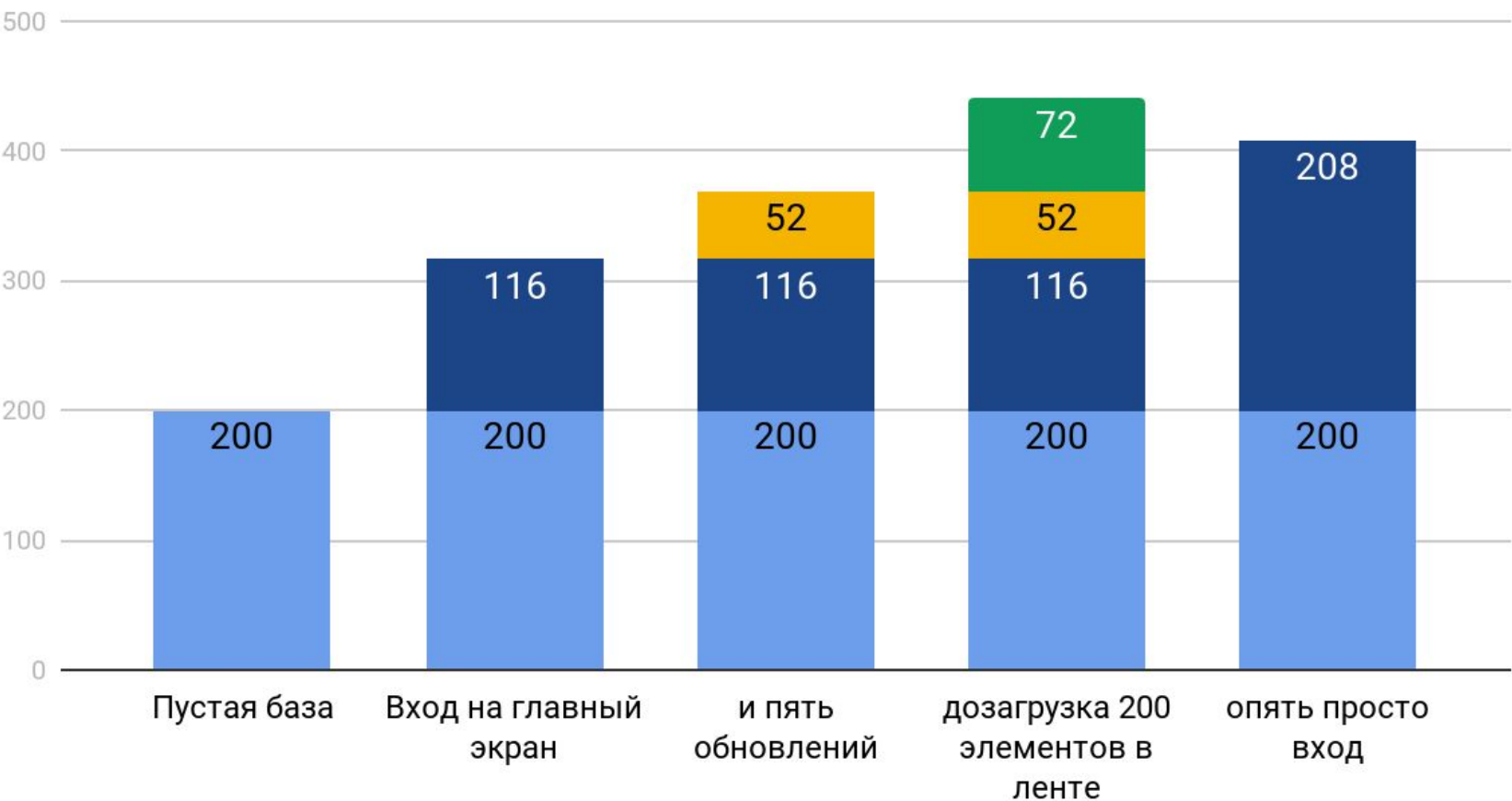
Пользовательский сценарий (Шаги)

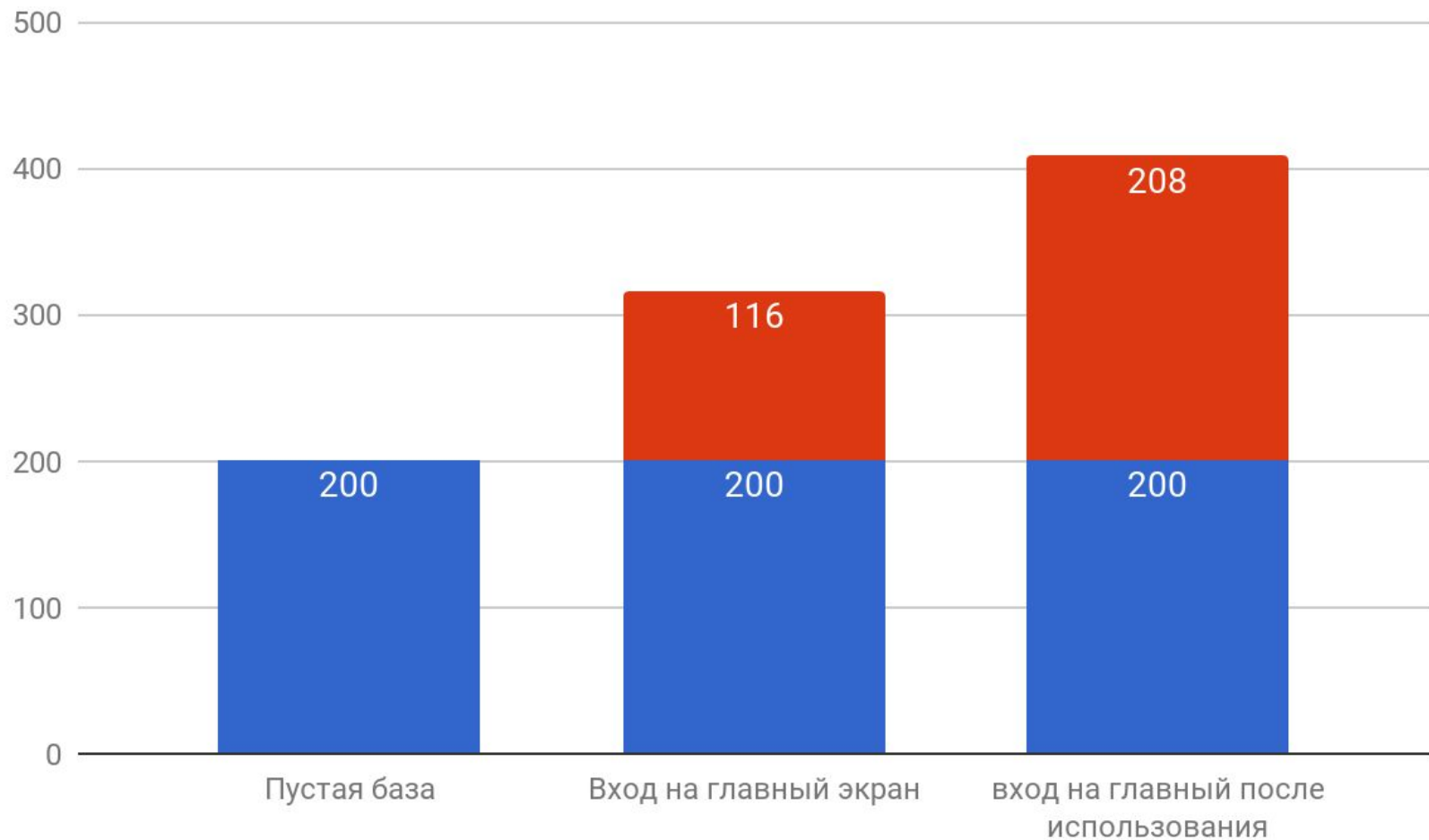


Мой сценарий

1. Авторизоваться
2. Войти на главный экран
3. Обновить главный экран
4. Догрузить элементы в списках на главном экране
5. Выйти до ввода пин-кода

Размер файла бд в кб

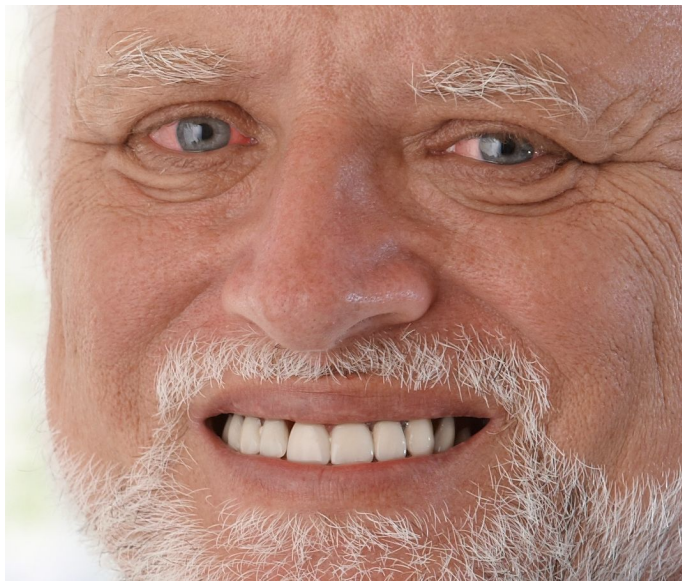




Что дальше?

Freelist pages

Debug



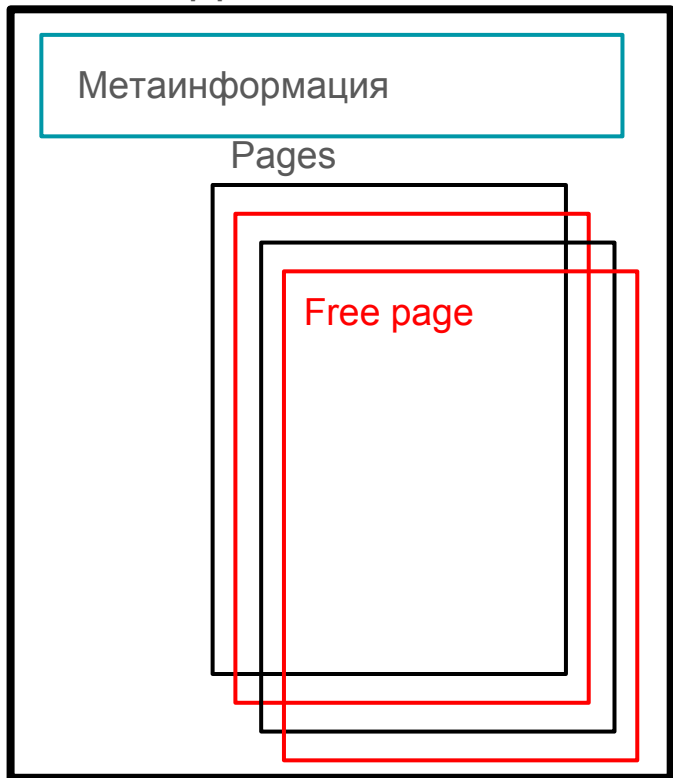
Freelist pages

<https://sqlite.org/fileformat2.html#freelist>

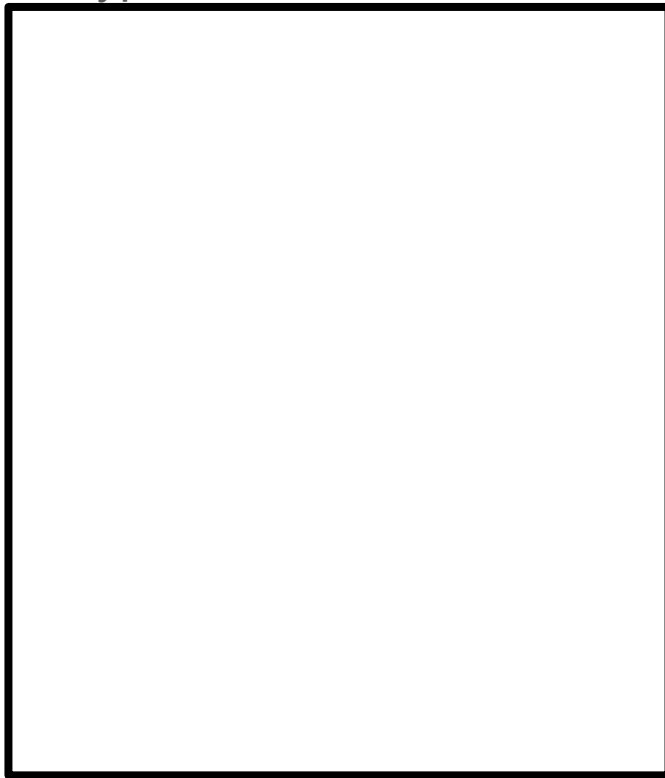
После удаления данные удаляются, но свободное место превращается в Freelist pages

Структура хранения SQLite

Файл БД



Журнал



Метаинформация

Всего 100 байт

С 32 по 36 байт номер первого Freelist trunk page

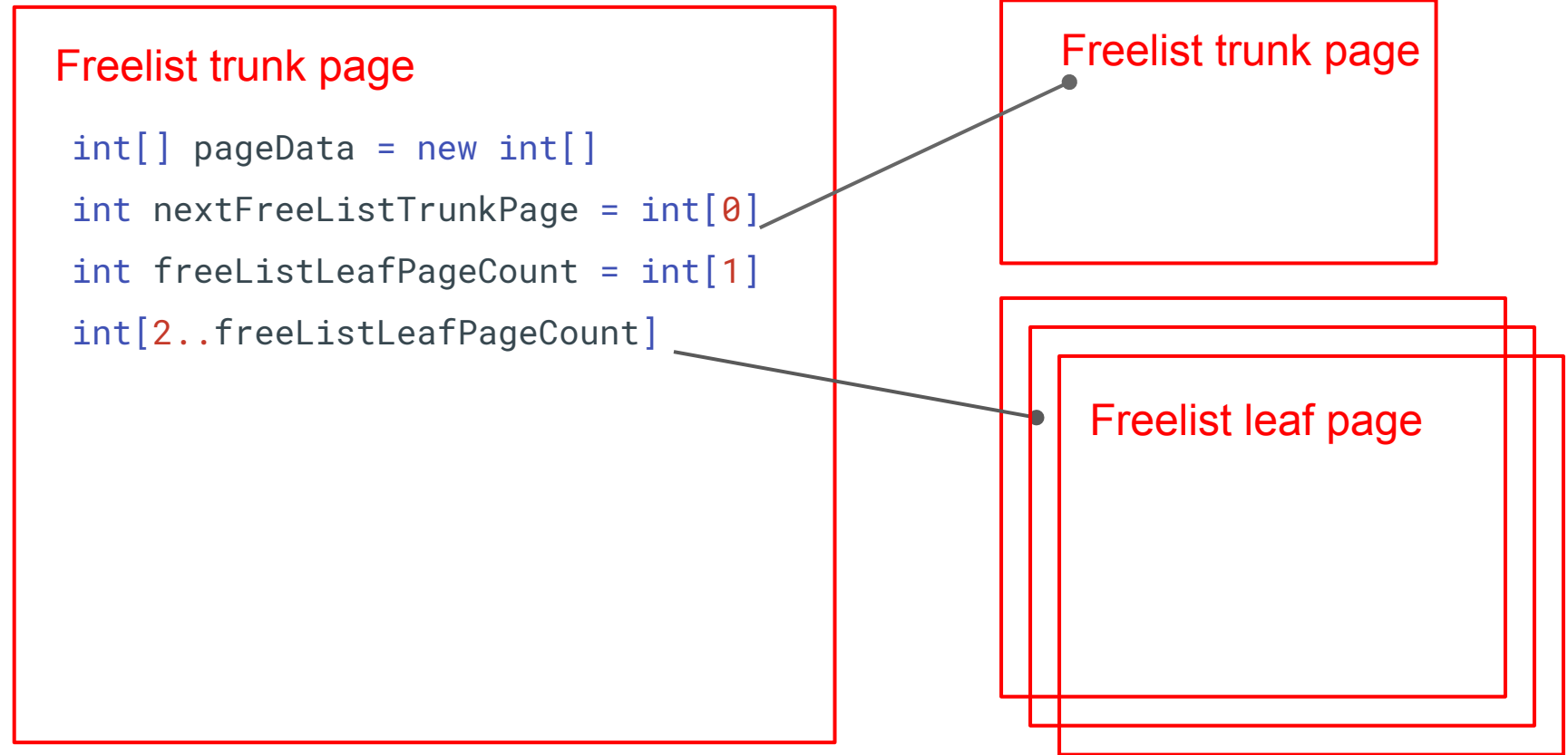
С 36 по 40 байт - кол-во Freelist pages

Структура Freelist trunk page

Freelist trunk page

```
int[] pageData = new int[]  
int nextFreeListTrunkPage = int[0]  
int freeListLeafPageCount = int[1]  
int[2..freeListLeafPageCount]
```

Freelist trunk page



Freelist leaf page

Как действительно уменьшить файл бд

Vacuum

Auto-vacuum

Vacuum

Перестройка файла бд

Изменение page-size, auto-vacuum

Auto-vacuum

```
PRAGMA schema.auto_vacuum = 0 | NONE | 1 | FULL | 2 | INCREMENTAL;
```

Vacuum

```
public void onOpen(SQLiteDatabase db) {  
    super.onOpen(db);  
    db.execSQL("VACUUM;");  
}
```

Debug

Логирование

Stetho

DbInspector

Логирование

OrmLite

adb shell setprop log.tag.StatementExecutor VERBOSE

adb shell setprop log.tag.BaseMappedStatement VERBOSE

adb shell setprop log.tag.MappedCreate VERBOSE

adb shell setprop log.tag.ORMLite DEBUG

adb shell setprop log.tag.droidDatabaseConnection VERBOSE

adb shell setprop log.tag.ndroidCompiledStatement VERBOSE

He OrmLite

adb shell setprop log.tag.SQLiteLog VERBOSE

adb shell setprop log.tag.SQLiteStatements VERBOSE

Stetho features

Network Inspections

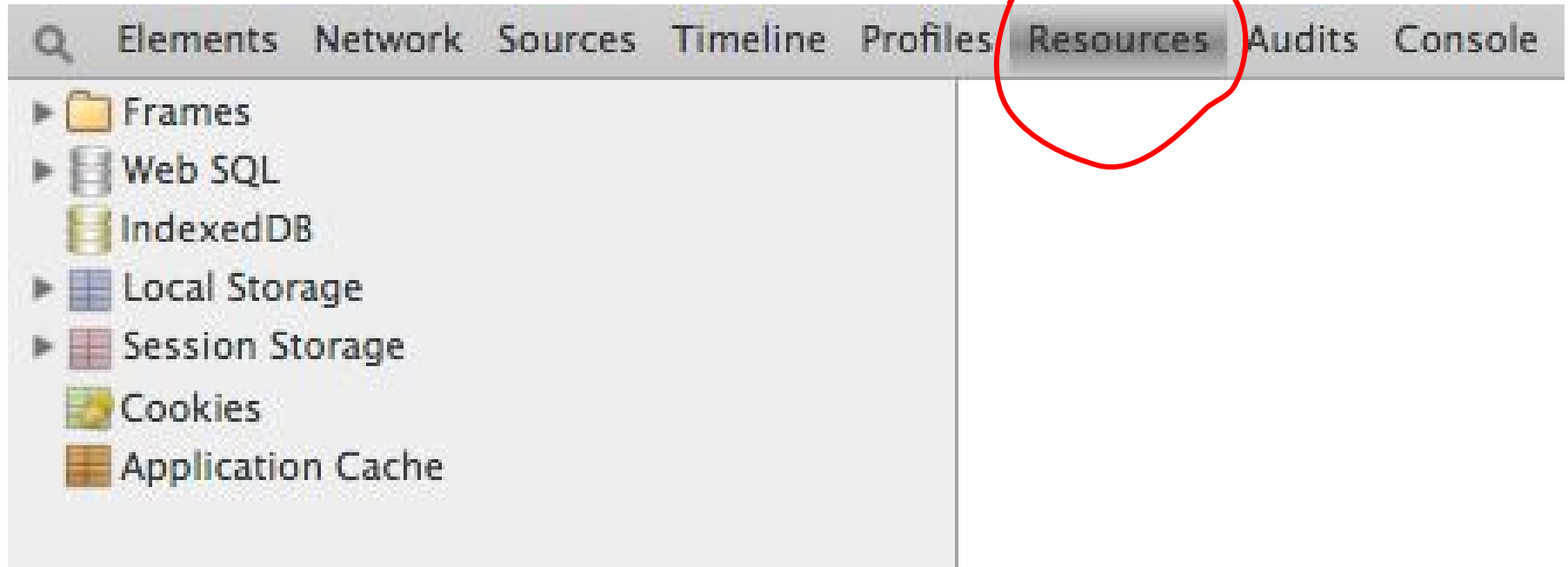
Local resource Inspection

View Hierarchy

Dumpapp

JavaScript Console

Stetho Resources Inspection



Stetho Resources Inspection

Elements	Network	Sources	Timeline	Profiles	Resources	Audits	Console
Frames							
Web SQL							
45882d71f444e001f6c045f40d6a2120							
account							
accountIntegration							
agreementStatistics							
android_metadata							
asset							
bank							
card							
cardValue							
counterparty							
CounterpartyVerification							
counterpartyInfo							
integration							
invoice							
InvoiceDocumentSourceInfo							
operation							
operationInfo							

1	1	TNC_129...	7000000...		1505254...	{blob}	1700	0	948e951c...	CONTRAG...	Debit
2	2	TNC_128...	7000000...		1505206...	{blob}	8800	0	b358172...	TAX	Debit
3	3	TNC_128...	7000000...		1505206...	{blob}	80	0	dc0b47...	BUDGET	Debit
4	4	TNC_128...	7000000...		1505206...	{blob}	1160	0	9e3cc060...	TAX	Debit
5	5	TNC_128...	7000000...		1505206...	{blob}	2040	0	9b8bcf00...	TAX	Debit
6	6	TNC_128...	7000000...		1505206...	{blob}	7550	0	8ce00949...	CONTRAG...	Debit
7	7	TNC_128...	7000000...		1505206...	{blob}	60284.89	0	d80a0ba...	CONTRAG...	Debit
8	8	TNC_129...	7000000...		1505163...	{blob}	35430.96	0			Credit
9	9	TNC_128...	7000000...		1505163...	{blob}	38692.39	0			Credit
10	10	AC_1300...	7000000...			{blob}	5500	0		REGULAR...	Debit
11	11	TNC_127...	7000000...		1505094...	{blob}	1000	0	63abceb3...	CONTRAG...	Debit
12	12	TNC_127...	7000000...		1505094...	{blob}	2700	0	23c9f105...	CONTRAG...	Debit
13	13	TNC_127...	7000000...		1505094...	{blob}	140414.02	0	5bdc5a27...	CONTRAG...	Debit
14	14	TNC_127...	7000000...		1505077...	{blob}	16615.68	0			Credit
15	15	TNC_127...	7000000...		1505077...	{blob}	1926.99	0			Credit
16	16	TNC_127...	7000000...		1505077...	{blob}	15888.89	0			Credit
17	17	TNC_126...	7000000...		1504818...	{blob}	27014.32	0			Credit
18	18	TNC_125...	7000000...		1504731...	{blob}	34817.99	0			Credit
19	19	TNC_124...	7000000...		1504645...	{blob}	28772.81	0			Credit
20	20	TNC_123...	7000000...		1504558...	{blob}	43506.13	0			Credit
21	21	TNC_121...	7000000...		1504488...	{blob}	5200	0	4035390...	TAX	Debit
22	22	TNC_122...	7000000...		1504472...	{blob}	17994.3	0			Credit
23	23	TNC_122...	7000000...		1504472...	{blob}	2047	0			Credit

С помощью Stetho

Я понял, что данные в связанных таблицах остаются

Связи между таблицами курильщика

```
CREATE TABLE artist(  
    artistid  INTEGER PRIMARY KEY,  
    artistname TEXT  
);
```

```
CREATE TABLE track(  
    trackid   INTEGER,  
    trackname TEXT,  
    trackartist INTEGER  
);
```

Связи между таблицами здорового человека

```
CREATE TABLE artist(  
  artistid  INTEGER PRIMARY KEY,  
  artistname TEXT  
);
```

```
CREATE TABLE track(  
  trackid   INTEGER,  
  trackname TEXT,  
  trackartist INTEGER,  
  FOREIGN KEY(trackartist) REFERENCES artist(artistid)  
);
```

Stetho foreign keys



The screenshot shows the Chrome DevTools Resources panel. The left sidebar lists 'Frames' and 'Web SQL'. Under 'Web SQL', a database with ID '68b6999abb715641e2af6dd323dff411' is selected, showing a table named 'account'. The main panel displays a SQL query: `PRAGMA foreign_key_list(account);`. Below the query, a table shows the result of the query.

id	seq	table
0	0	paymentID

`PRAGMA foreign_key_list(table_name);`

Результат

table	from	to	on_update	on_delete
paymentID	accountID	id	NO ACTION	CASCADE

DbInspector

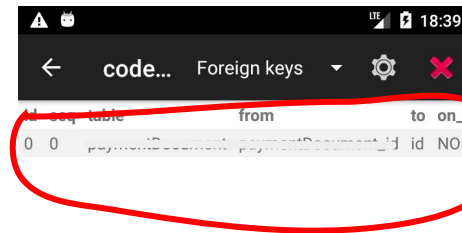
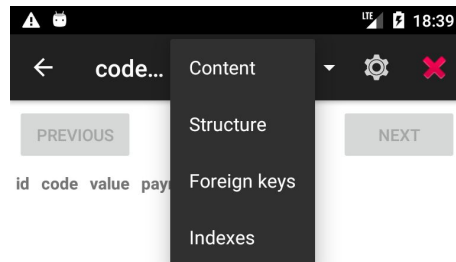
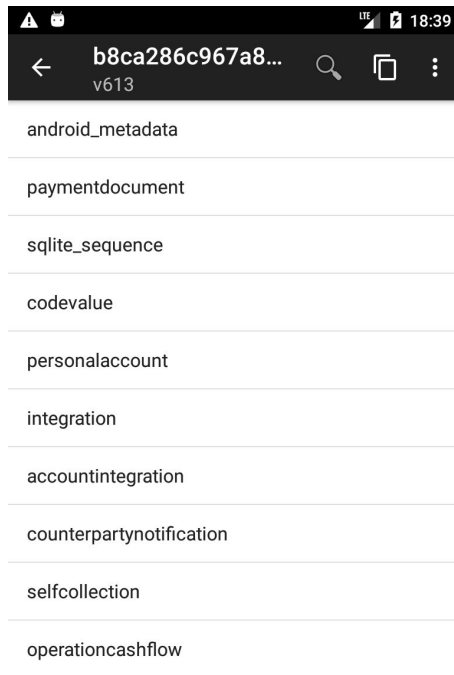
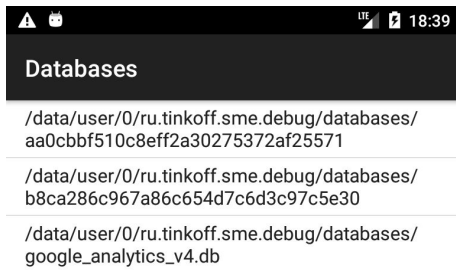
Content

Structure

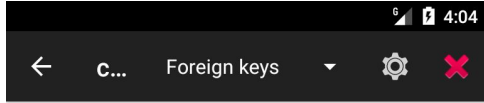
Foreign Keys

Indexes

DbInspector foreign keys



Foreign keys in PaymentDocument



_ (ツ) _/

Foreign связей нет

Потому что проблемы у нас в коде

Потому что проблемы где-то еще...



Foreign keys y hac

@DatabaseTable

```
public class PaymentDocument {
```

```
    @DatabaseField(foreign = true,  
                    foreignAutoRefresh = true,  
                    foreignAutoCreate = true)
```

```
    private ShortCounterParty counterParty;
```

```
}
```


Delete PaymentDocuments

```
public void deleteByEq(String field, Object value) throws SQLException {  
    DeleteBuilder builder = deleteBuilder();  
    builder.setWhere(builder.where().eq(field, value));  
    builder.delete();  
}
```

Проблем в коде нет, что дальше?

Goog|еить

Яндексить

Duckduckgoить

etc...

Foreign key constraints in SQLite (it is disabled by default)



Привет, android foreign keys

В SQLite с [version 3.6.19](#) (14-10-2009)

[https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#setForeignKeyConstraintsEnabled\(boolean\)](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#setForeignKeyConstraintsEnabled(boolean)) - min Api 16

Pragma for rescue Foreign Constraints

@Override

```
public void onOpen(SQLiteDatabase db) {  
    super.onOpen(db);  
    if (!db.isReadOnly()) {  
        // Enable foreign key constraints  
        db.execSQL("PRAGMA foreign_keys=ON;");  
    }  
}
```

Foreign keys - появились, но код сломался

Хранить данные как blob

ON DELETE, ON UPDATE Actions

Foreign keys to BLOB

Если дочерний объект ссылается одна таблица

Если дочерний объект не участвует в выборке

Если нам не нужен lazy fetching

Если нам не нужны миграции

Foreign keys to BLOB

```
@DatabaseField(dataType = DataType.SERIALIZABLE)
```

```
private ShortCounterParty counterParty;
```

Foreign keys + ON Update, ON Delete Actions

No Action

Restrict

Set Null

Set Default

Cascade

NO ACTION

```
CREATE TABLE track(  
    trackid    INTEGER,  
    trackname  TEXT,  
    trackartist INTEGER,  
    FOREIGN KEY(trackartist) REFERENCES artist(id)  
);
```

RESTRICT

```
CREATE TABLE track(  
    trackid    INTEGER,  
    trackname  TEXT,  
    trackartist INTEGER REFERENCES artist(id) ON DELETE RESTRICT,  
);
```

SET NULL

```
CREATE TABLE track(  
    trackid    INTEGER,  
    trackname  TEXT,  
    trackartist INTEGER REFERENCES artist(id) ON DELETE SET NULL,  
);
```

SET DEFAULT

```
CREATE TABLE track(  
    trackid    INTEGER,  
    trackname  TEXT,  
    trackartist INTEGER DEFAULT 0 REFERENCES artist(id) ON DELETE SET  
DEFAULT  
);
```

CASCADE

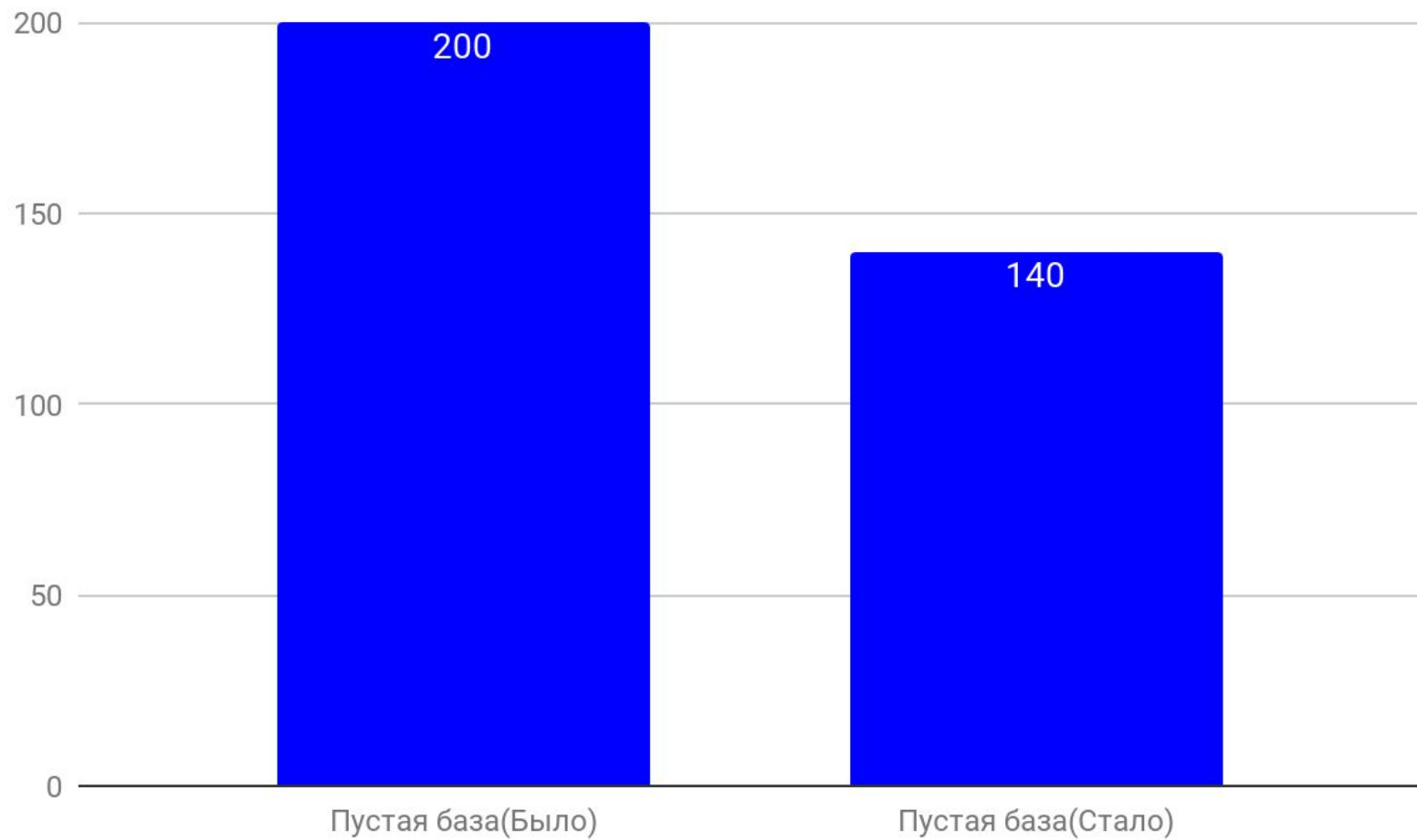
```
CREATE TABLE track(  
    trackid    INTEGER,  
    trackname  TEXT,  
    trackartist INTEGER REFERENCES artist(id) ON DELETE CASCADE  
);
```

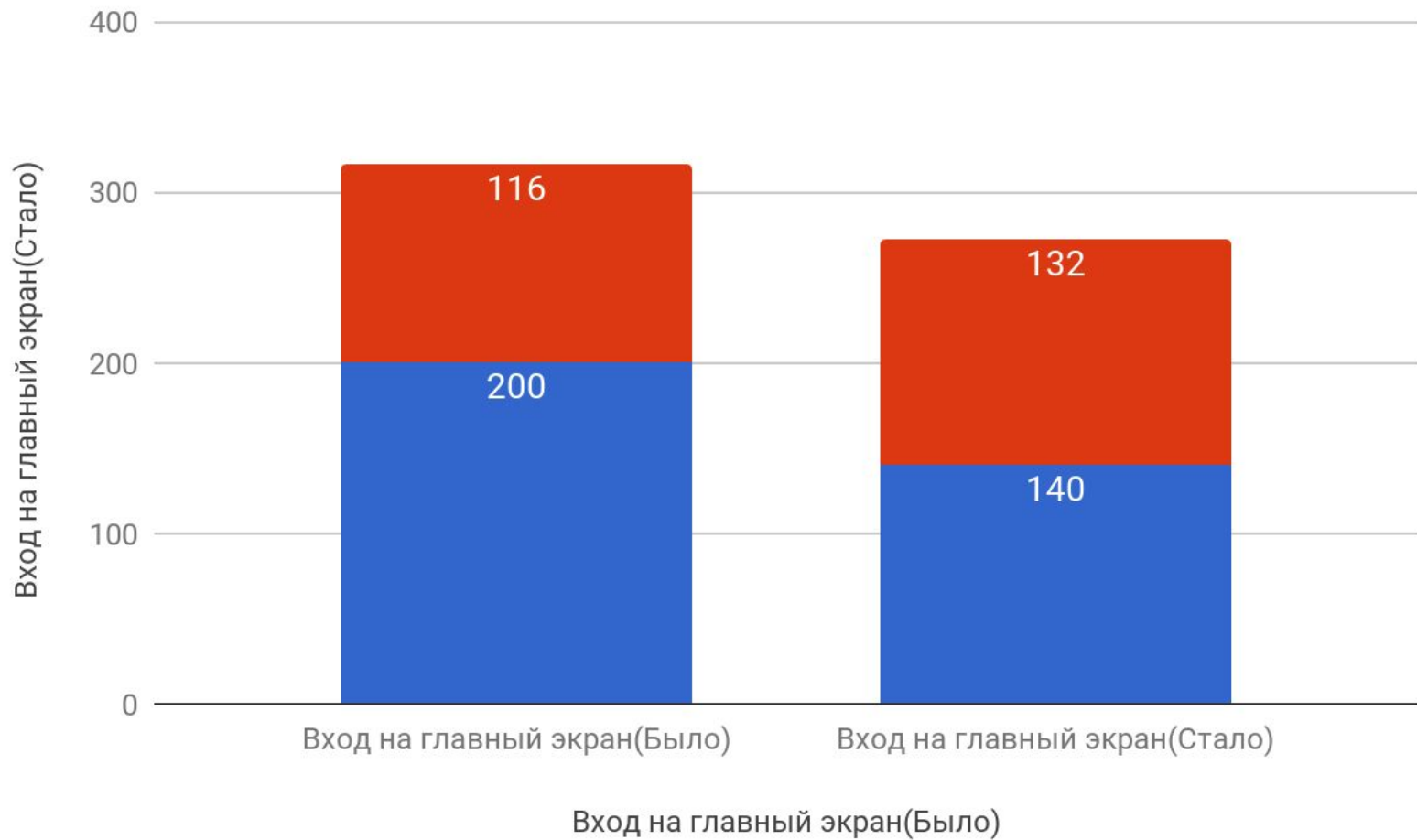

Foreign keys + ON UPDATE ON Delete Actions

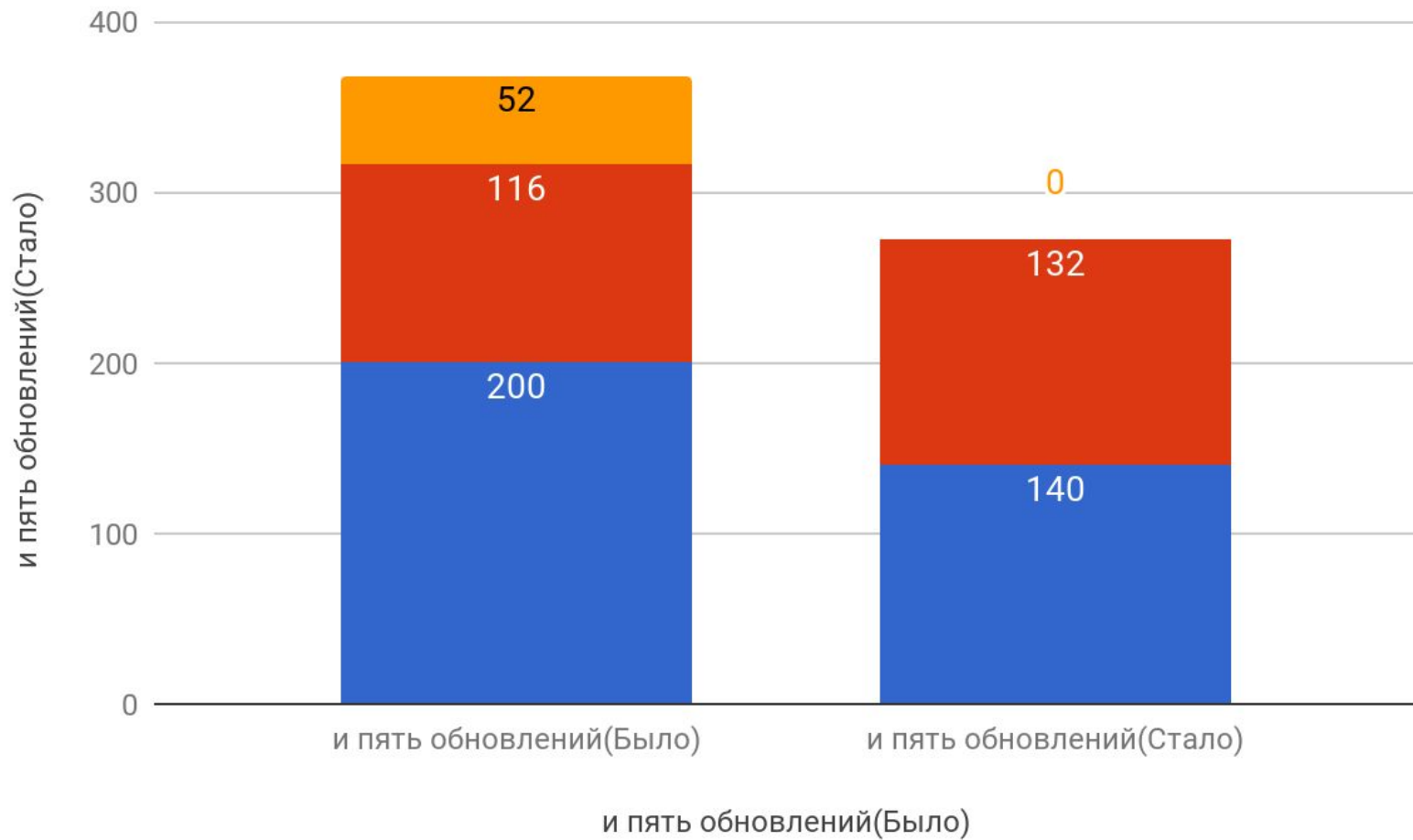
```
@DatabaseField(foreign = true,  
    columnDefinition =  
        "integer references paymentDocument (id) on delete cascade")  
private PaymentDocument paymentDocument;
```

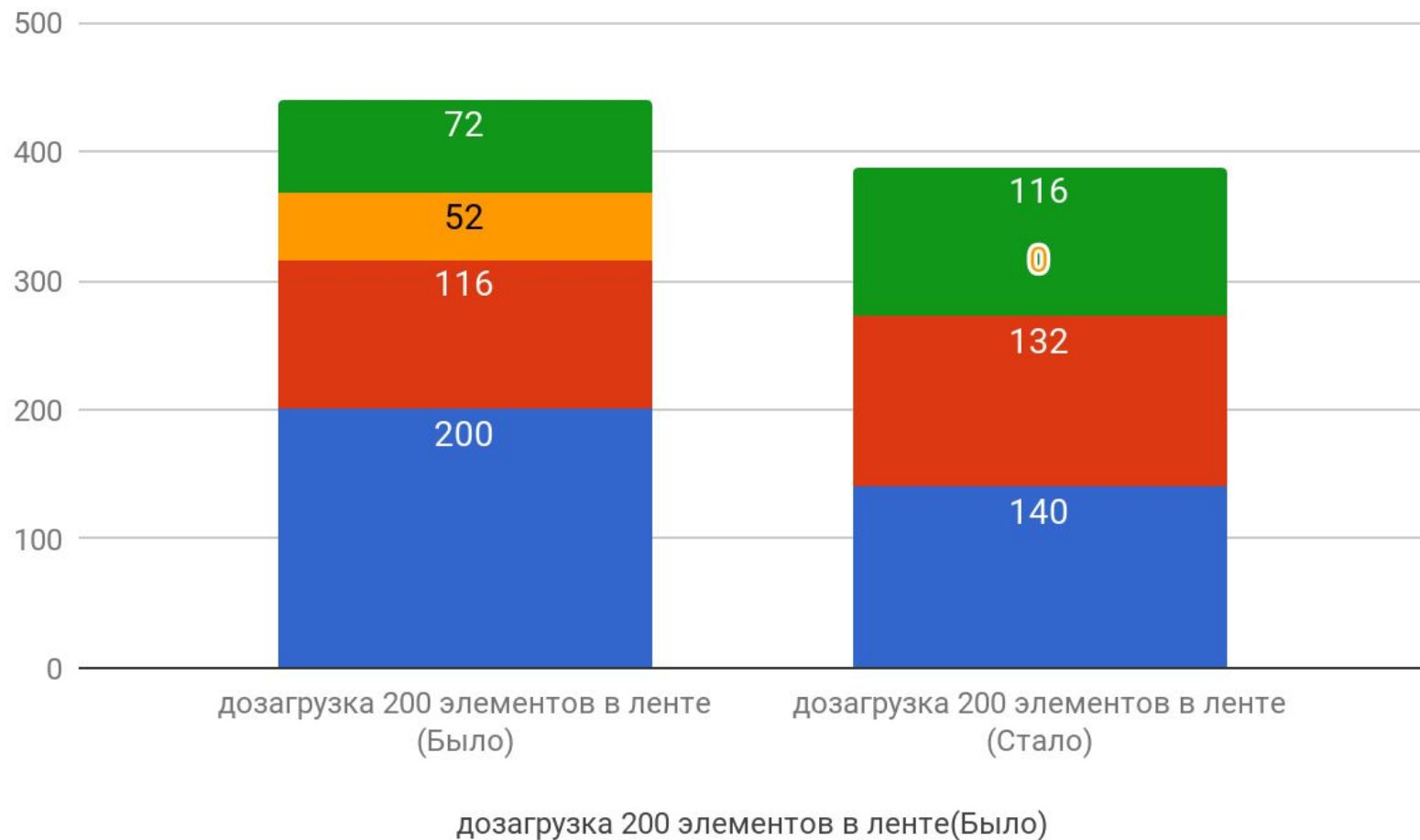
Foreign keys with DROP TABLE

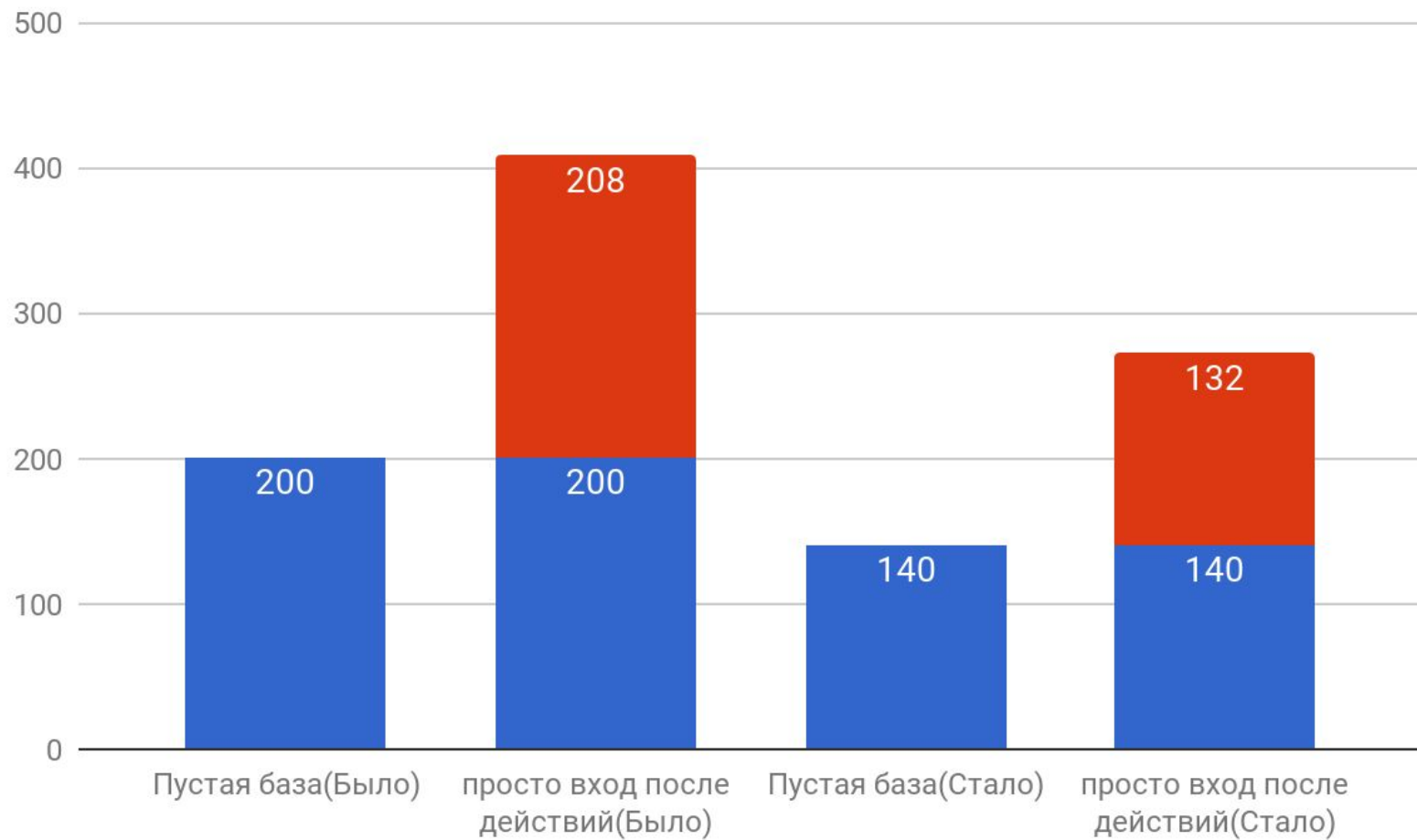
```
public void deleteAll() {  
    TableUtils.dropTable(dbHelper.getConnectionSource(), clazz, true);  
    TableUtils.createTable(dbHelper.getConnectionSource(), clazz);  
}
```











Выводы

Не думайте, что у вас проблема с Freelist page

Узнайте лучше свою любимую Orm

Пишите тесты на слой бд

Подключайте тулинг на ранних этапах

LOONEY TUNES



"That's all Folks!"

Ссылки

<http://hibernate.org/orm/what-is-an-orm/>

<http://ormlite.com/>

<https://stackoverflow.com/questions/7255574/reached-max-size-for-compiled-sql-statement-cache-for-database>

<http://www.sqlite.org/pragma.html>

<http://facebook.github.io/stetho/>

[https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#setForeignKeyConstraintsEnabled\(boolean\)](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#setForeignKeyConstraintsEnabled(boolean))

https://sqlite.org/lang_vacuum.html

<https://sqlite.org/fileformat2.html#freelist>