

SQLite

android



Способы доступа СУБД к БД

- **Файл-серверные** (Microsoft Access, FoxPro)
- **Клиент-серверные** (Oracle, SQL Server, Firebird, DB2, PostgreSQL, MySQL)
- **Встраиваемые** (SQLite, Realm)

Introduction of New Database Technologies 1994–2014

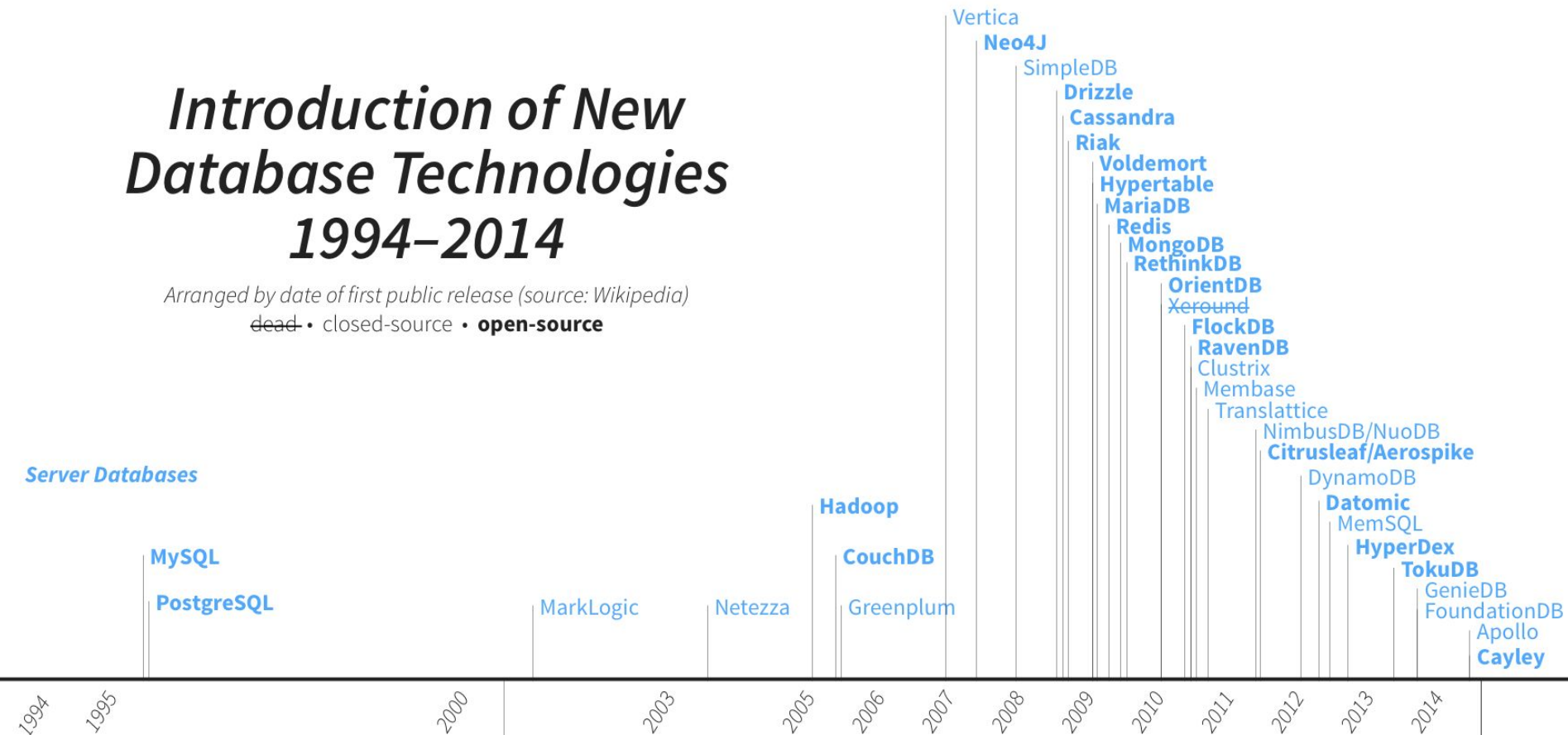
Arranged by date of first public release (source: Wikipedia)

~~dead~~ • closed-source • open-source

Server Databases

Mobile Databases

Realm



Встраиваемые СУБД

Встраиваемая система управления базами данных — архитектура систем управления базами данных, когда СУБД тесно связана с программой, и работает на том же устройстве, не требуя профессионального администрирования. Встраиваемые СУБД применяются во многих программах, которые хранят большие объёмы данных, но при этом не требуется доступ со многих компьютеров.

Примеры использования

Программы, в которых может быть использована встраиваемая СУБД: почтовые клиенты и мессенджеры (базы переписки), медиапроигрыватели (плей-листы и обложки), просмотрщики изображений (метаданные и уменьшенные эскизы), различные локальные БД наподобие телефонных справочников и геоинформационных систем.

https://en.wikipedia.org/wiki/Embedded_database

SQLite – компактная встраиваемая реляционная система управления базами данных. «Встраиваемая» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций API библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том же устройстве, на котором исполняется программа.

Почему именно SQLite?

- **Минимальные затраты ресурсов.** Для работы большинства СУБД необходим специальный процесс сервера базы данных. SQLite обходится без сервера, база данных SQLite представляет собой обычный файл. Когда база данных не используется, она не расходует процессорное время. Это особенно важно на мобильных устройствах, чтобы избежать разрядки аккумулятора.
- **Оптимизация для одного пользователя.** С базой данных взаимодействует только наше приложение, поэтому можно обойтись **без идентификации с именем пользователя и паролем**.
- **Надежность и быстрота.** SQLite поддерживает транзакции - если при обновлении нескольких блоков данных что-то пойдет не так, SQLite сможет вернуться к исходному состоянию. Кроме того, операции чтения и записи данных реализуются на оптимизированном коде C.

Где хранится БД?

Android автоматически создаёт для каждого приложения папку, в которой хранятся базы данных этого приложения, например `/data/data/com.sunmeat.thirdproject/databases`. В этой папке приложение может хранить несколько баз данных. Каждая БД состоит из двух файлов. Имя первого — файла базы данных — соответствует имени БД: например, «**dating**». Это основной файл баз данных SQLite, в нём хранятся все данные. Второй файл — файл журнала. Его имя состоит из имени базы данных и суффикса «-journal» — например, «**dating-journal**». В файле журнала хранится информация обо всех изменениях, внесённых в БД. Если вдруг в работе с данными возникнет проблема, Android использует данные журнала для отмены (или отката) последних изменений.

Как выглядит основной файл

```
Step — Блокнот
Файл  Правка  Формат  Вид  Справка
SQLiteformat3+  @←|γ^|  ←-в#^ЛиИ§×:ЛиР^--|+
+ Ytablessqlite_sequencesqlite_sequence|CREATETABLEsqlite_sequence
(name,seq)к1•| ́)tableStudentsStudents^CREATETABLEStudents
(idINTEGERPRIMARYKEYAUTOINCREMENT,nameTEXTNOTNULL)W  -|--  ctablea
ndroid_metadataandroid_metadata^CREATETABLEandroid_metadata
(localeTEXT)  ×ч×ч•  γ|ru_RU↑Л,×й×Ц×Б×И×>×€×s×d×M×:×
%×ТЛяЛМЛЧЛИЛЛ±ЛћЛ%ЛzЛсЛРЛ;Л,↑^!Р•Р»РμPSP°!!|  ^-Р'Р°Р»РμPSC,РёPS◀^L)РќР
°С,Р°Р»СЪСЦ^L^L1РЎС,Р°PSPёСГР»Р°PI||^!Р•Р»РμPSP°!!!^L-Р'Р°Р»РμPSC
,РёPS◀^L)РќР°С,Р°Р»СЪСЦ^L^L1РЎС,Р°PSPёСГР»Р°PI+^!Р•Р»РμPSP°!!×^L-Р'Р
°Р»РμPSC,РёPS◀Л)РќР°С,Р°Р»СЪСЦ^L^L1РЎС,Р°PSPёСГР»Р°PI?^!Р•Р»РμPSP
°!!^L-Р'Р°Р»РμPSC,РёPS◀^L)РќР°С,Р°Р»СЪСЦ^  ^L1РЎС,Р°PSPёСГР»Р°PI■^L!
Р•Р»РμPSP°!!•^L-Р'Р°Р»РμPSC,РёPS◀^L)РќР°С,Р°Р»СЪСЦ^|  ^L1РЎС,Р
°PSPёСГР»Р°PI^L^!Р•Р»РμPSP°!!^L-Р'Р°Р»РμPSC,РёPS◀^L)РќР°С,Р
°Р»СЪСЦ^  ^L1РЎС,Р°PSPёСГР»Р°PI  ×Т×Т?  ^  Students↑|
```

Классы для работы с SQLite

- Помощник SQLite – создаётся расширением класса **SQLiteOpenHelper**. Он предоставляет средства для создания и управления базами данных.
- Класс базы данных **SQLiteDatabase** – предоставляет доступ к БД.
- Класс **Cursor** – предназначен для чтения и записи в базу данных.

Безопасность данных

- **Если при подключении к базе данных не указывается имя пользователя и пароль, то как обеспечивается безопасность данных?**
- Каталог, в котором хранятся базы данных приложения, доступен для чтения только для самого приложения. Безопасность доступа к базе данных обеспечивается на уровне операционной системы.

Внешние БД

- **Возможно ли написать приложение Android, которое работает с внешней базой данных — например, SQL Server?**
- Ничто не мешает подключаться к другим БД по сети, но не стоит забывать об экономии ресурсов, используемых Android. Например, обращение к базе данных через веб-службу может более экономно расходовать заряд аккумулятора.

Обновление приложения

- **Папка базы данных находится в папке приложения?**
- Нет. База данных хранится в другой папке, отдельно от кода приложения. Это позволяет установить обновлённую версию приложения без потери информации в базе данных.

Задачи помощника SQLite

- **Создание базы данных** – при первой установке приложения файл базы данных не существует. Помощник SQLite следит за тем, чтобы файл базы данных был создан с правильным именем и с правильной структурой таблиц.
- **Обеспечение доступа к базе данных** – приложению не обязательно знать все подробности о том, где хранится файл базы данных. Помощник SQLite предоставляет удобный объект, представляющий базу данных, и приложение работает с базой через этот объект – тогда, когда сочтёт нужным.
- **Сопровождение баз данных** – может случиться так, что структура базы данных изменится со временем. Тогда помощник преобразует старую версию в новую, с учётом всех последних изменений.

Типы данных SQLite

SQLite поддерживает типы TEXT (аналог String в Java), INTEGER (аналог long в Java) и REAL (аналог double в Java). Остальные типы следует конвертировать, прежде чем сохранять в базе данных. SQLite сама по себе не проверяет типы данных, поэтому вы можете записать целое число в колонку, предназначенную для строк, и наоборот.

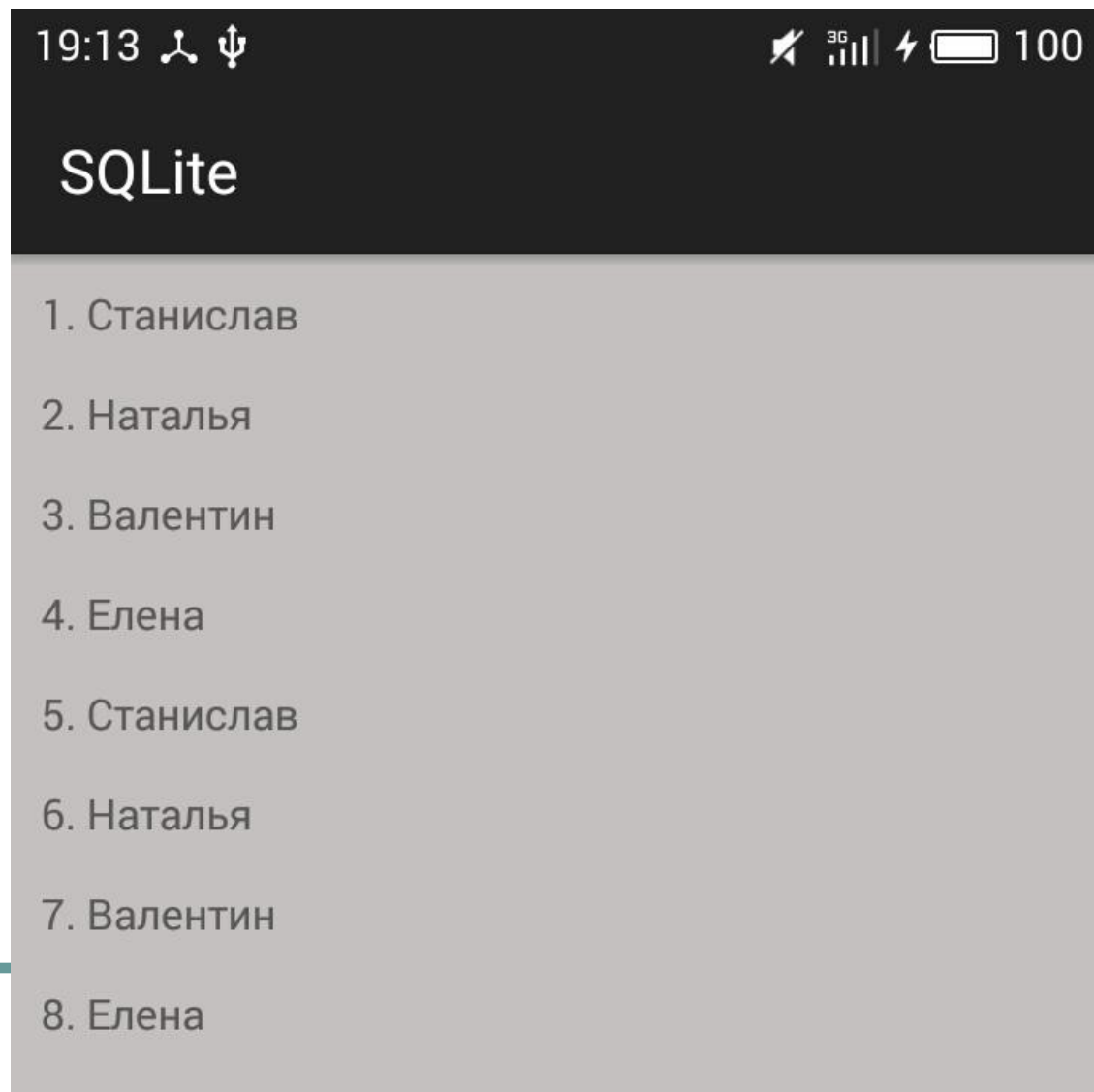
Тип	Описание
NULL	пустое значение
INTEGER	целочисленное значение
REAL	значение с плавающей точкой
TEXT	строки или символы в кодировке UTF-8, UTF-16BE или UTF-16LE
NUMERIC	здесь можно хранить булевы значения, а также время и дату
BLOB	бинарные данные

Первая база (одна таблица)

- Создание БД и одной таблицы
- Добавление записей (**insert**)
- Выборка записей (**query**)
- Обновление и удаление
- Получение количества записей
- Исходники:

<https://git.io/vXcgO>

Результаты



Практика

Добавить в класс Student такие поля:

- Фамилия
- Дата рождения (TEXT)
- Время добавления строки в базу
- Средний балл (REAL)







Оформить дополнительные аксессоры и конструкторы.

Вторая база (многотабличная)

- Реализация связи «многие-ко-многим»
- Создание таблицы для задач
- Создание таблицы тегов (меток)
- Добавление, обновление и удаление
- Исходники:

<https://git.io/vXcpv>

Результаты

21:29     3G   100

SQLite

Количество тегов: 4
Количество задач: 10

Названия тегов: Покупки, Важно, Посмотреть, Работа,

Названия задач: новый комп, модный шарф, протеин, Doctor Strange, Resident Evil: The Final Chapter, American Crime Story, позвонить маме, сходить в качалку, сделать презентацию, пойти на корпоратив,

Что я собрался посмотреть: Doctor Strange, Resident Evil: The Final Chapter, American Crime Story,

Было заданий: 10

Не пойду на корпоратив. Осталось заданий: 9

Деньги закончились - а значит, никаких покупок!
Осталось заданий: 6

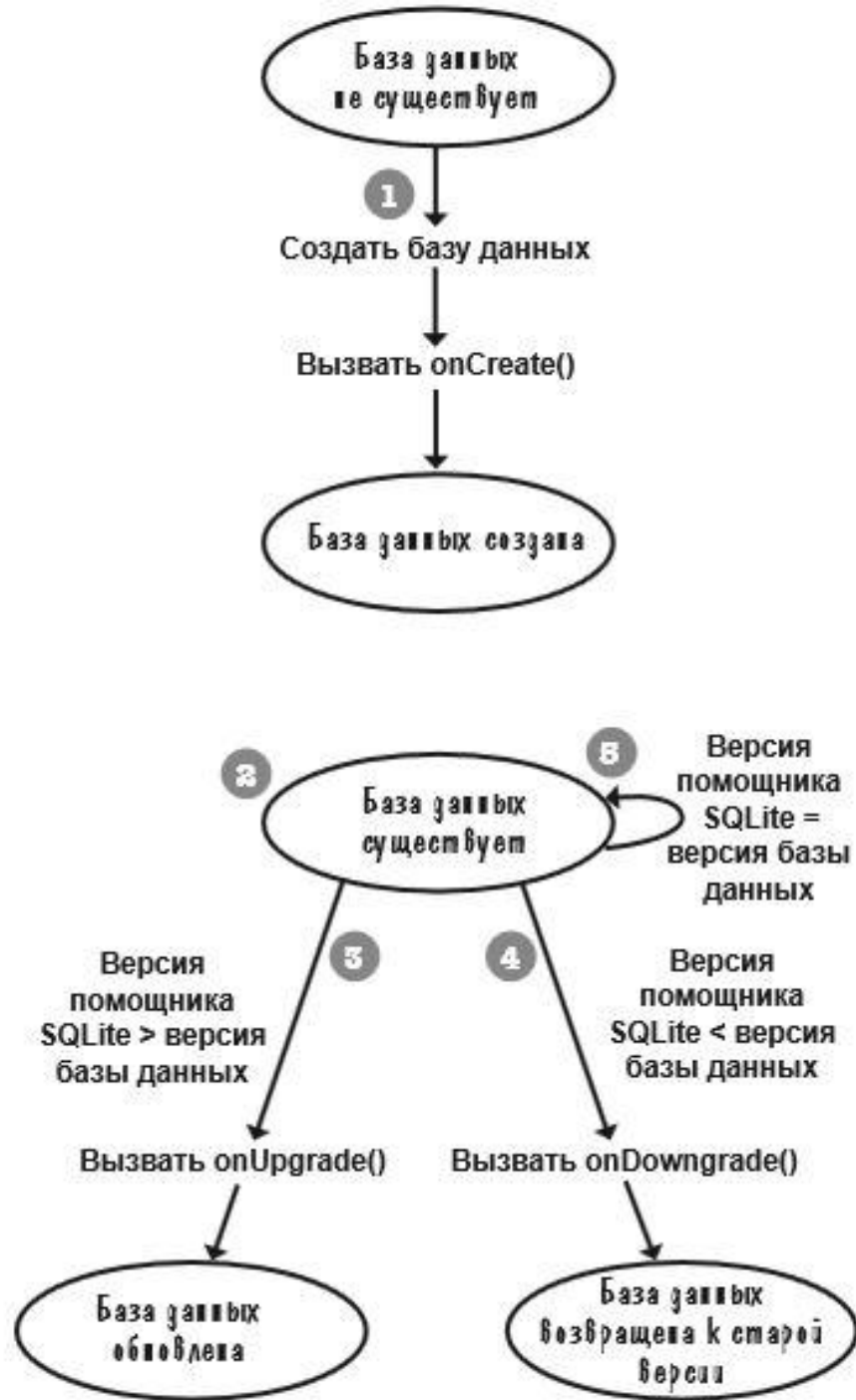
Практика

В задание со студентом добавить таблицу Группы (обновить БД до версии 2).
Программно добавить несколько групп, назначить студентам номера групп.
Программно удалить группу: если группы ещё есть, перевести студентов из удаляемой группы в первую из оставшихся групп, а если групп больше нет – то удалить студентов вместе с группой.

Подключение к БД

Итак, в примерах было показано, что при подключении к БД нужно указать **название** БД и **версию**. При этом могут возникнуть следующие ситуации:

- **БД не существует.** Это может быть, например, в случае первичной установки программы. В этом случае приложение должно **создать** БД и все таблицы в ней. И далее оно уже работает с только что созданной БД.
- **БД существует**, но её версия **устарела**. Это может быть в случае обновления программы. Например, новой версии программы нужны дополнительные поля в старых таблицах или новые таблицы. В этом случае приложение должно **апдейтить** существующие таблицы и создать новые, если это необходимо.
- **БД существует** и её версия **актуальна**. В этом случае приложение успешно **подключается** к БД и работает с ней.



1 Если база данных не существует, то помощник SQLite создает базу данных, после чего выполняется метод помощника `onCreate()`.

2 Если база данных уже существует, то помощник SQLite сравнивает номер версии, хранящийся в базе данных, с номером версии в коде помощника.

3 Если номер версии в коде помощника SQLite больше номера версии базы данных, вызывается метод `onUpgrade()`. После обновления помощник SQLite изменяет номер версии базы данных.

4 Если номер версии в коде помощника SQLite меньше номера версии базы данных, вызывается метод `onDowngrade()`. После выполнения возврата помощник SQLite изменяет номер версии базы данных.

5 Если номер версии в коде помощника SQLite равен номеру версии базы данных, не вызывается ни один из методов. База данных уже находится в актуальном состоянии.

Дополнительные примеры

- Группировка и сортировка:
<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/76-urok-36-sqlite-podrobnee-pro-metod-query-uslovie-sortirovka-gruppirovka.html>
- Применение INNER JOIN:
<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/77-urok-37-zaprosy-iz-svjazannyh-tablits-inner-join-v-sqlite-metod-rawquery.html>
- Транзакции в SQLite:
<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/78-urok-38-tranzaktsii-v-sqlite.html>
- Хранение картинок в БД:
<http://developer.alexanderklimov.ru/android/sqlite/android-sqlite.php>
- Использование CursorLoader:
<http://startandroid.ru/ru/uroki/vse-uroki-spiskom/278-urok-136-c>

Что почитать про SQLite

- <http://www.vogella.com/tutorials/AndroidSQLite/article.html>
- <http://developer.alexanderklimov.ru/android/sqlite/android-sqlite.php>
- <http://startandroid.ru/ru/uroki/vse-uroki-spisokom/74-urok-34-hranenie-dannyh-sqlite.html>
- <https://developer.android.com/training/basics/data-storage/databases.html>