

GitHub

This repository ▾

Search or type a command



Explore

Features

Enterprise

Blog

Sign up

Sign in



realm / realm-cocoa

★ Star

1,540

🔗 Fork

85

Realm is a mobile database: a replacement for Core Data & SQLite <http://realm.io>

commits

🔗 54 branches

🏷️ 50 releases

tutors



realm

93 from

ed 6 minute

e6.xcodepr

ings

ing nil obje

ings

docs

- update docs readme

12 days ago

examples

Update Swift examples

9 days ago

plugin

- use placeholder in class templates

12 days ago

scripts

Don't add a duplicate copy of libtightdb-ios.a to the framework

14 days ago

tools

Version bump

a day ago

.dir-locals.el

Project specific Emacs settings updated: Always show trailing white-s...

11 months ago

.gitignore

The core folder is now a symlink to a versioned folder

24 days ago

<> Code

🔔 Issues

22

~ Puls

📊 Gra

HTTPS clone URL

<https://github.com/realm/>

You can clone with [HTTPS](#) or [Subversion](#). ?

🖥️ Clone in Desktop

📄 Download ZIP

What is Realm?

- **Fast, embedded database** (zero-copy, not an ORM)
- **Used in apps with *millions* of users**
- **NoSQL**
- **Full ACID transactions**
- **Well defined threading model**
- **Cross-platform C++ core with many language bindings**
(only Objective-C & Swift released)

Open Source*



github.com/realm/realm-cocoa

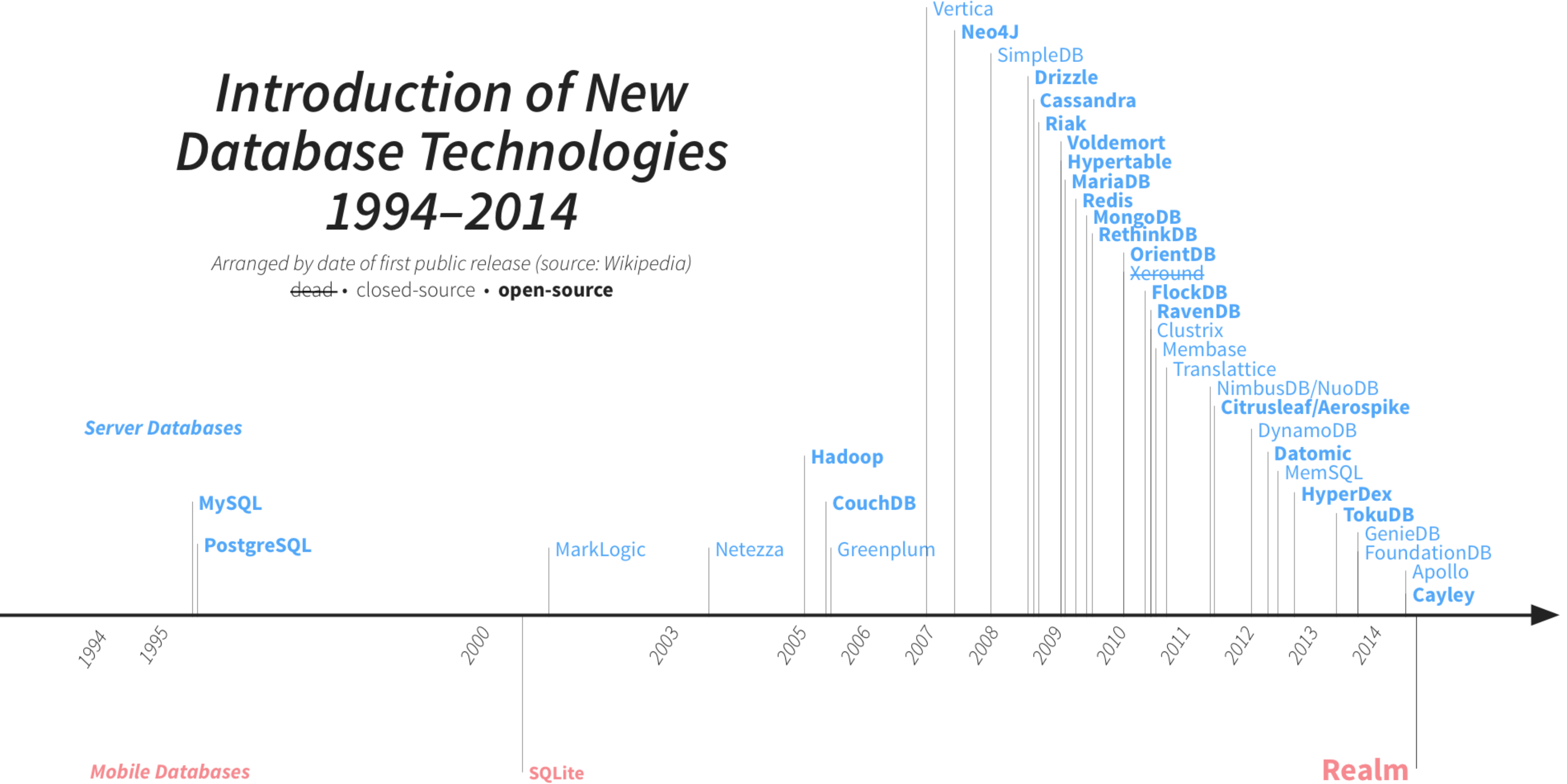
* Bindings 100% open source, C++ core being released as
Apache 2

Why?

Introduction of New Database Technologies 1994–2014


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

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



Current State of iOS Persistence

Core Data

- Full-featured & mature
- 10+ years old ORM for SQLite
- Slow
- Complex and difficult to learn/debug
-  only

SQLite

- Faster than Core Data
- Optimized for iOS
- Cross Platform
- 14+ years old
- Bad user experience
- Manual mapping and queries
- Lack of thread safety

Other Options

- FMDB
- YapDatabase
- CouchbaseLite
- LevelDB

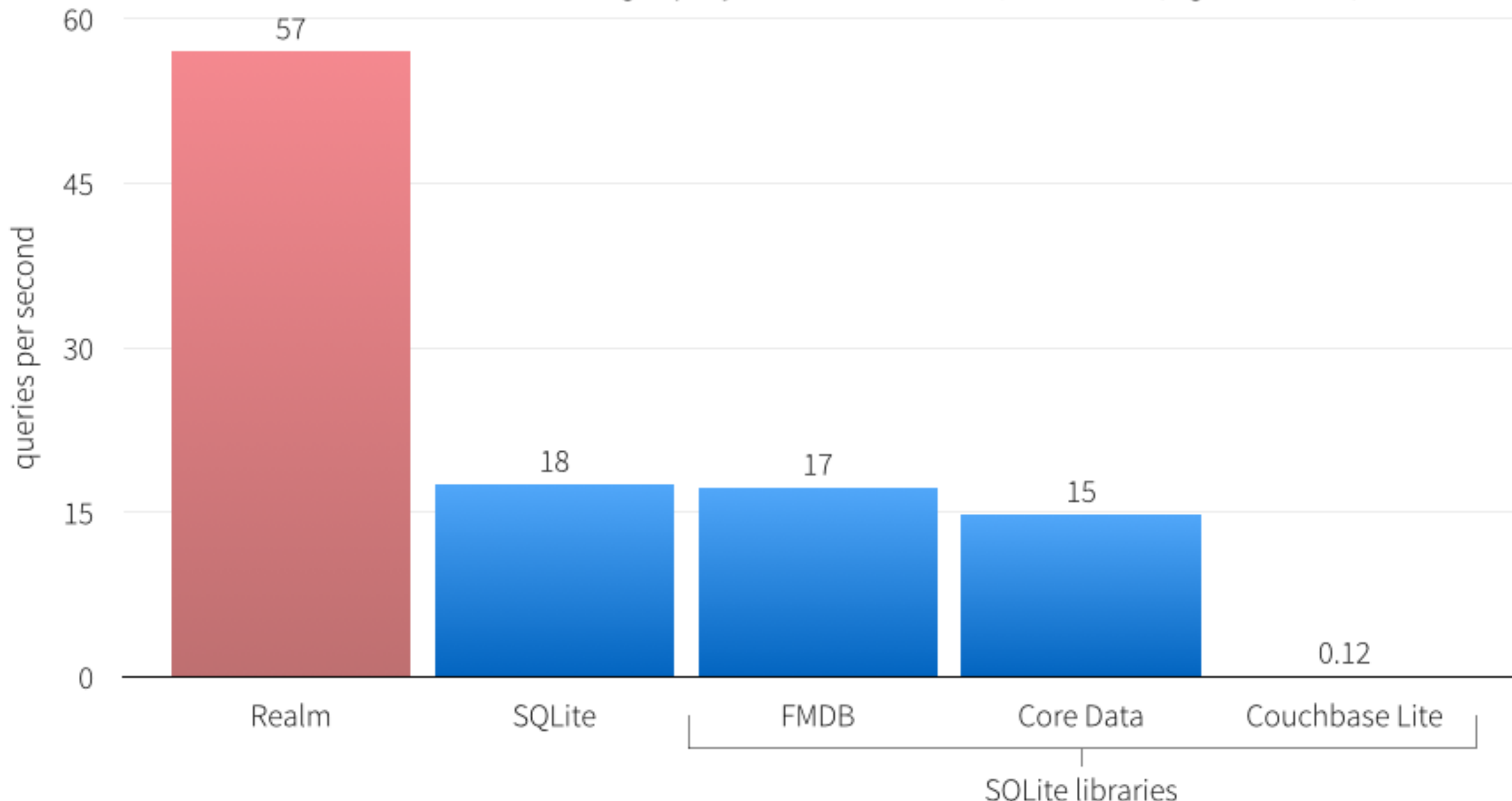
Lots has changed in last decade

- Smartphone Revolution
- NoSQL
- Need for Sync

Benchmarks

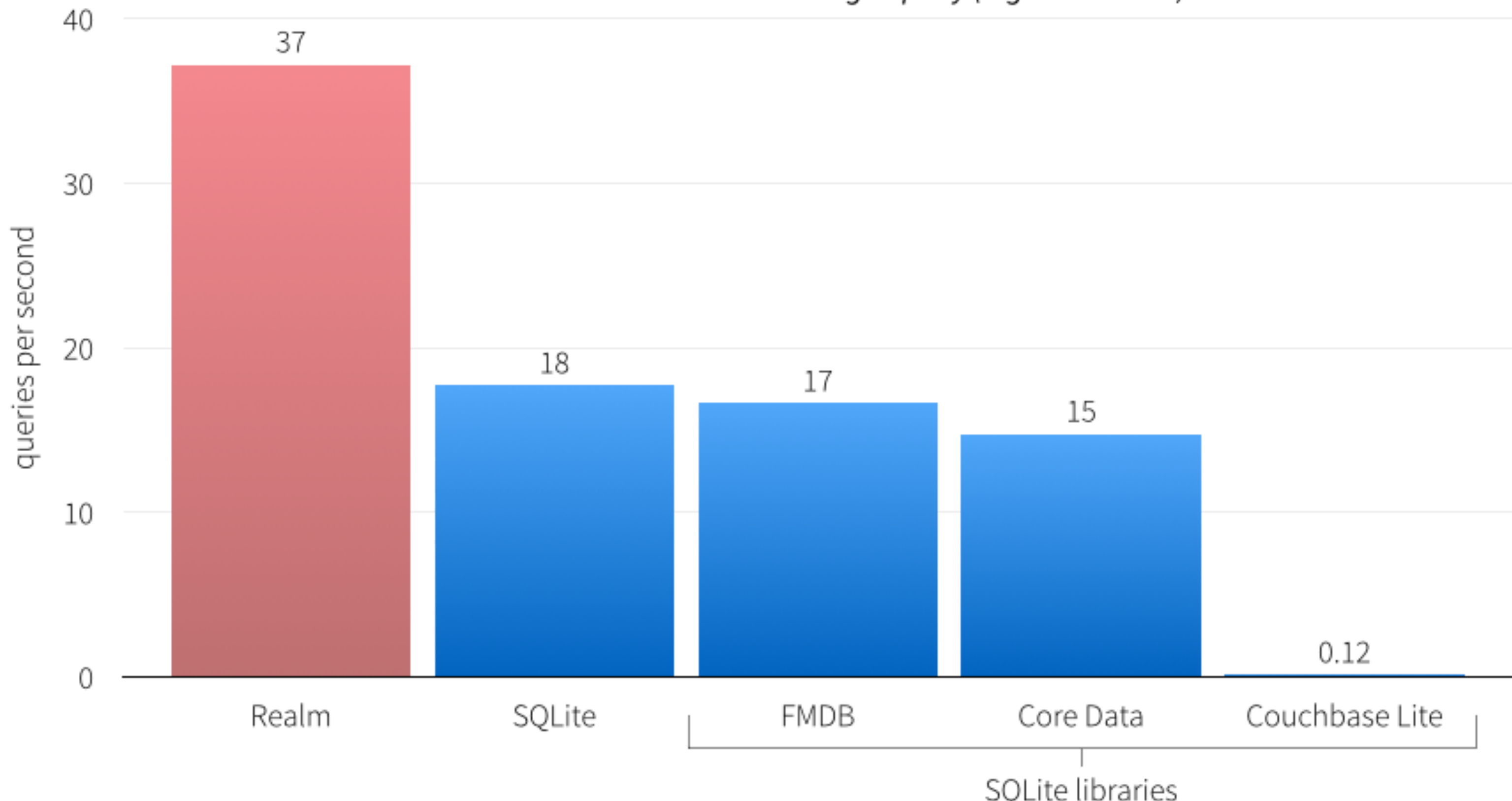
Counts

Get count of records matching a query on a database of 150,000 records (higher is better)



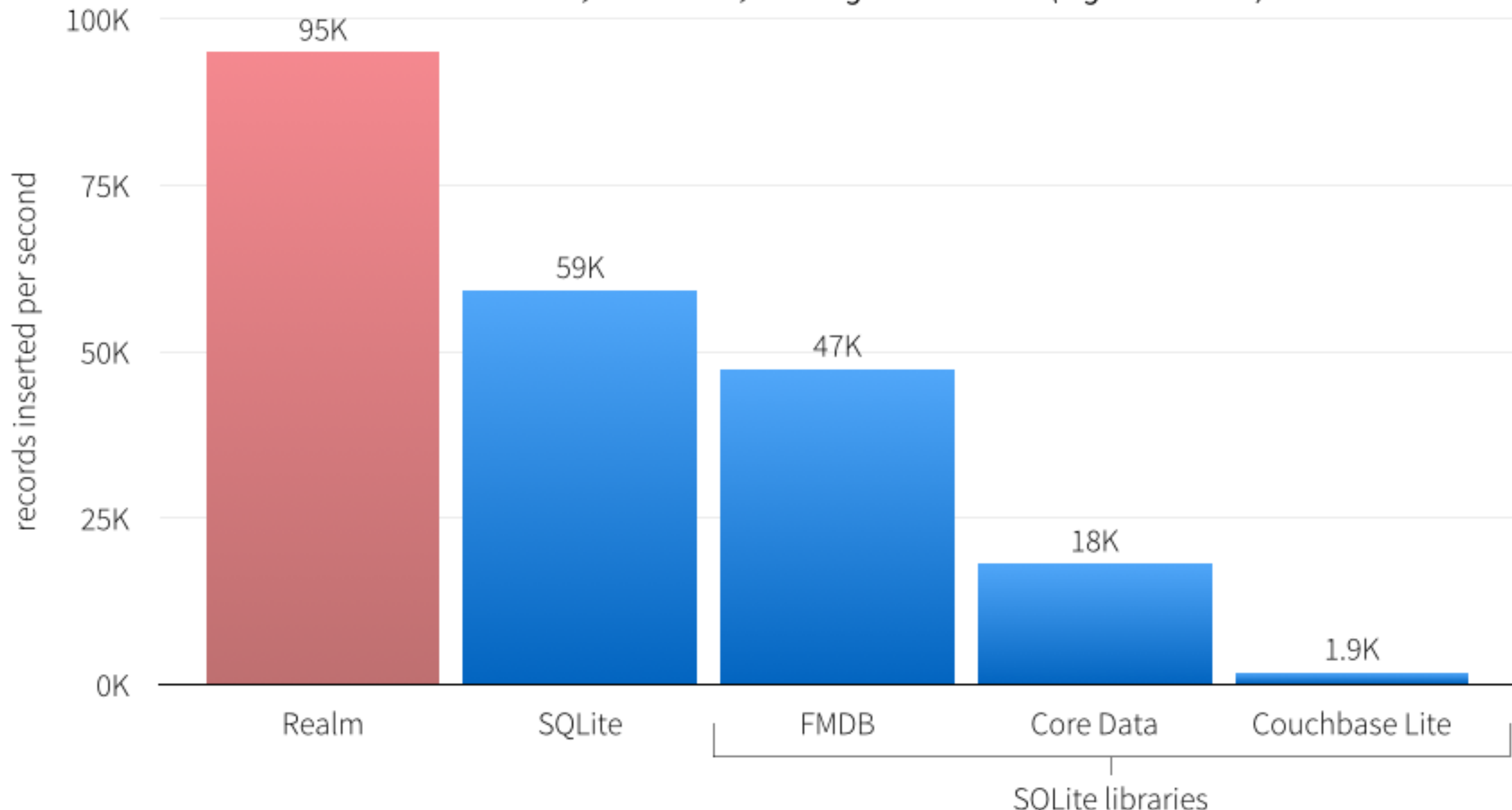
Queries

Iterate over all records matching a query (higher is better)



Inserts

Insert 150,000 records, in a single transaction (higher is better)



Realm Models

```
@interface Employee : RLMObject
@property NSString *name;
@property NSDate *startDate;
@property float salary;
@property BOOL fullTime;
@end
```

```
RLM_ARRAY_TYPE(Employee)
```

```
@interface Company : RLMObject
@property NSString *name;
@property Employee *ceo;
@property RLMArray<Employee> *employees;
@end
```

Realm Models (Swift)

```
class Employee: Object {  
    dynamic var name = ""  
    dynamic var startDate = NSDate()  
    dynamic var salary = 0.0  
    dynamic var fullTime = true  
}
```

```
class Company: Object {  
    dynamic var name = ""  
    dynamic var ceo = Employee()  
    var employees = List<Employee>()  
}
```


Using Realm

```
// Using Realm Objects
```

```
Company *company = [[Company alloc] init];  
company.name = @"Realm"; // etc...
```

```
// Transactions
```

```
RLMRealm *realm = [RLMRealm defaultRealm];  
[realm transactionWithBlock:^(  
    [realm addObject:company];  
)];
```

```
// Querying objects
```

```
RLMArray *companies = [Company allObjects];  
RLMArray *FTEmployees = [Employee objectsWhere:@"fullTime == YES"];
```

Using Realm (Swift)

```
// Using Realm Objects
```

```
let company = Company()
```

```
company.name = "Realm" // etc...
```

```
// Transactions
```

```
defaultRealm().write {
```

```
    realm.add(company)
```

```
}
```

```
// Querying objects
```

```
let companies = objects(Company)
```

```
companies[0]?.name // => Realm (generics)
```

```
let ftJohns = objects(Employee).filter(.fullTime == true && .name == "John")
```

BETA

Demo

XCODE

Work In Progress

- Change notifications/Live Results Sets
- Delete Rules
- Bi-directional relationships
- Sync
- Open Source Core
- Android

Questions?

@simjp, jp@realm.io