

What is Realm?

- **Fast, zero-copy, embedded database**
- **Used in apps with *millions* of users**
- **NoSQL**
- **Full ACID transactions**
- **Well defined threading model**
- **Cross-platform C++ core with many language bindings**
(currently Objective-C, Swift & Android)

Open Source*



github.com/realm/realm-cocoa

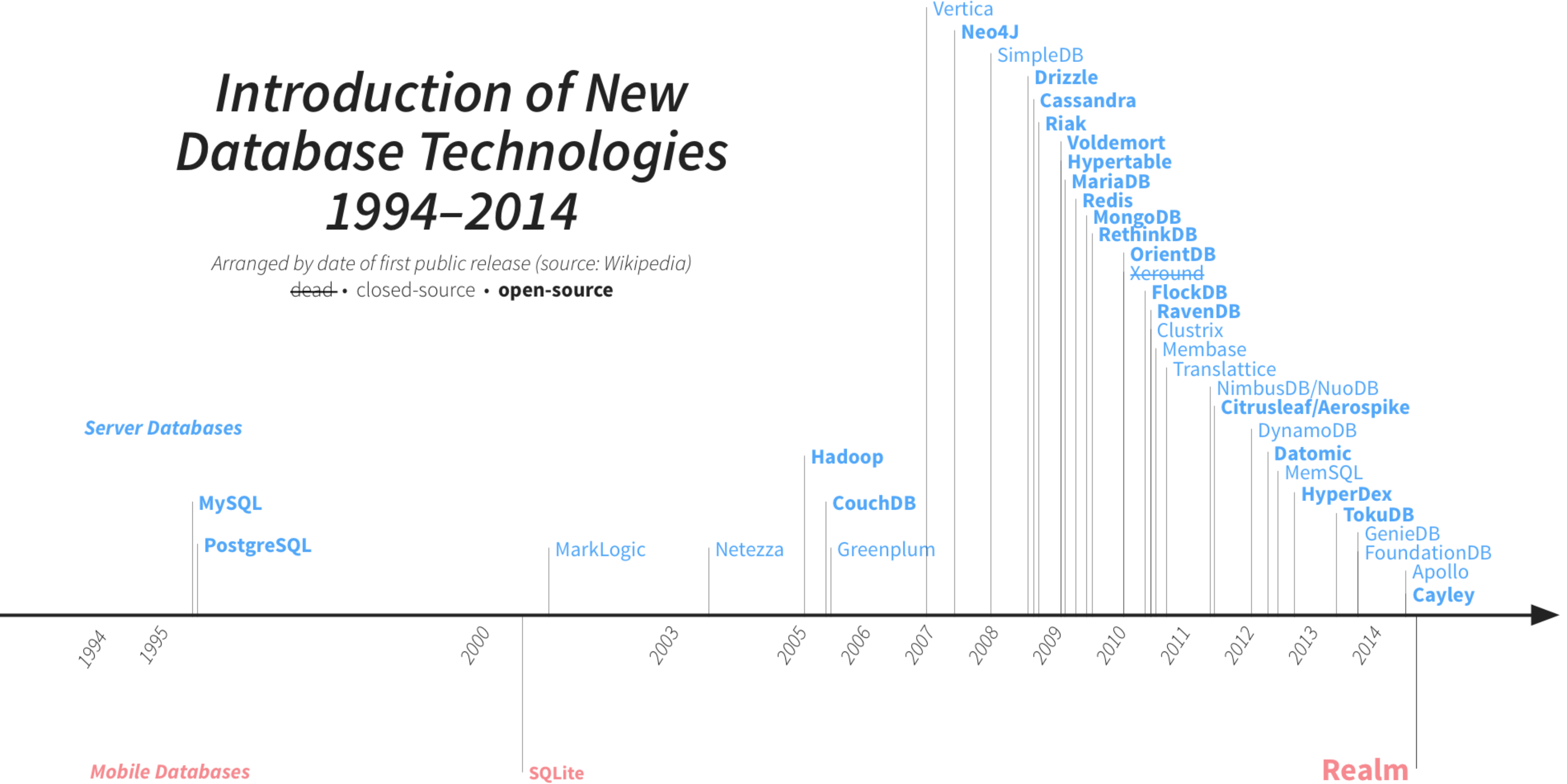
* Bindings 100% open source, C++ core launching as Apache 2

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

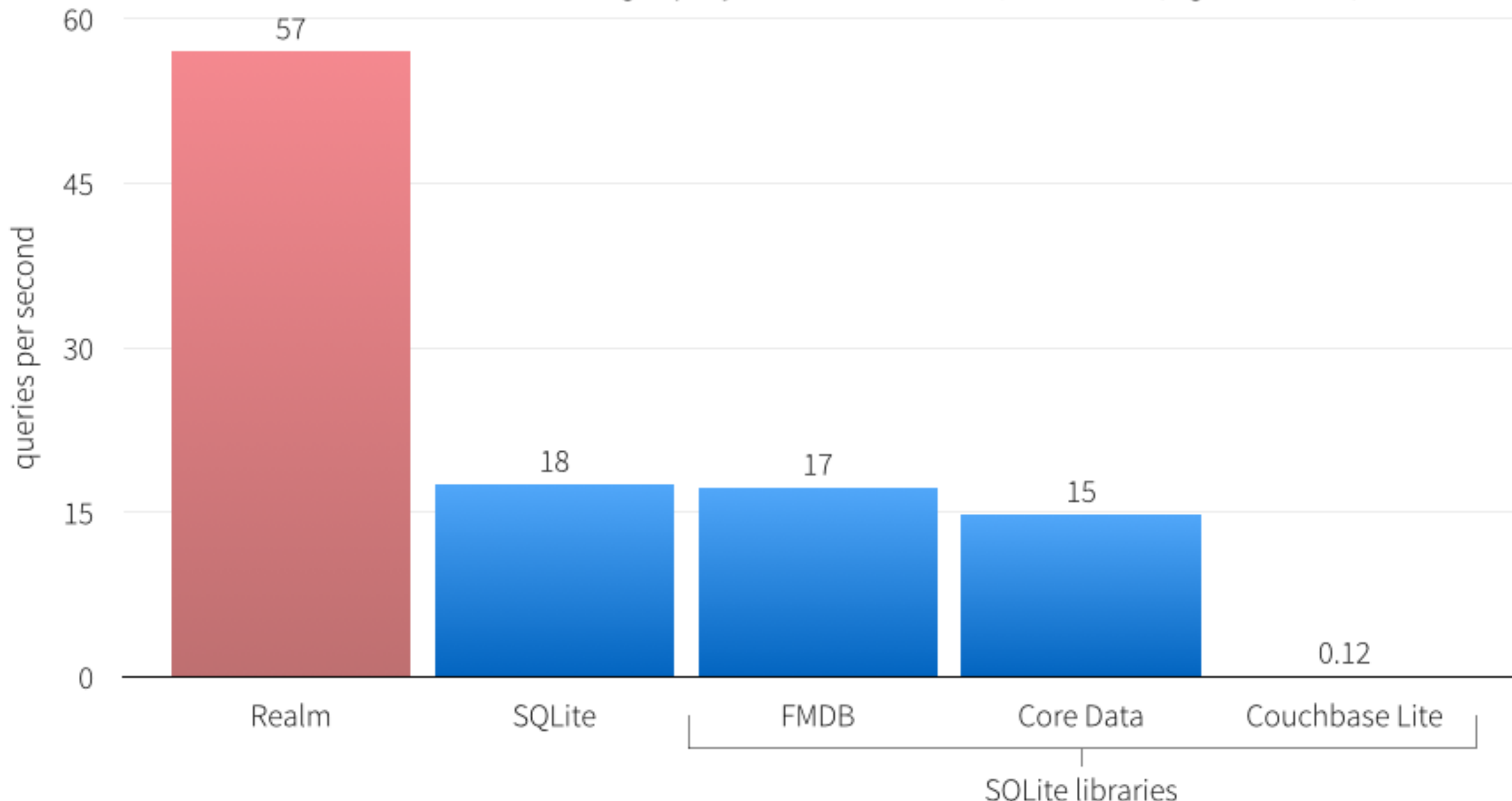
Lots has changed in last decade

- Smartphone Revolution
- NoSQL
- Need for Sync

Benchmarks

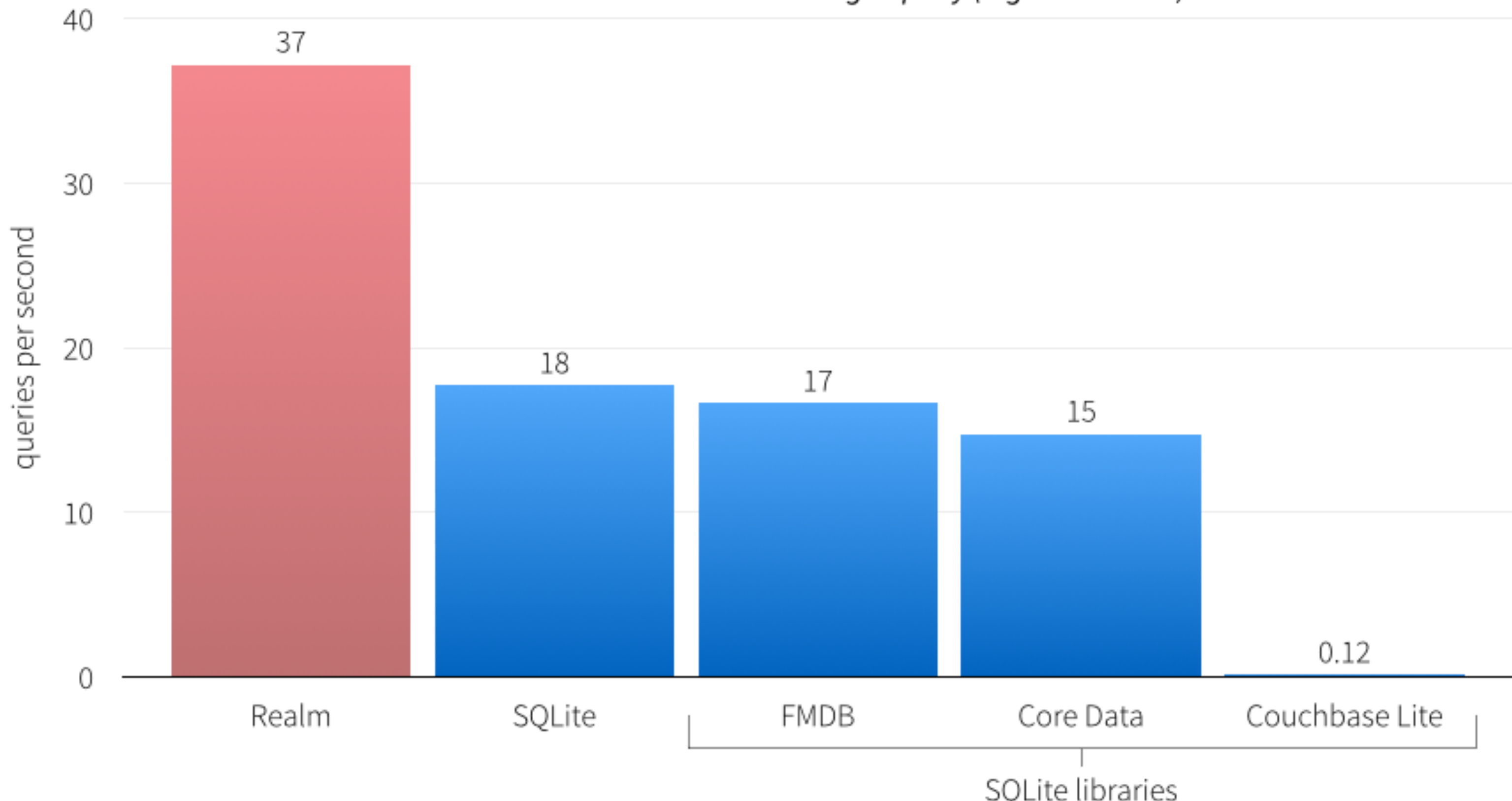
Counts

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



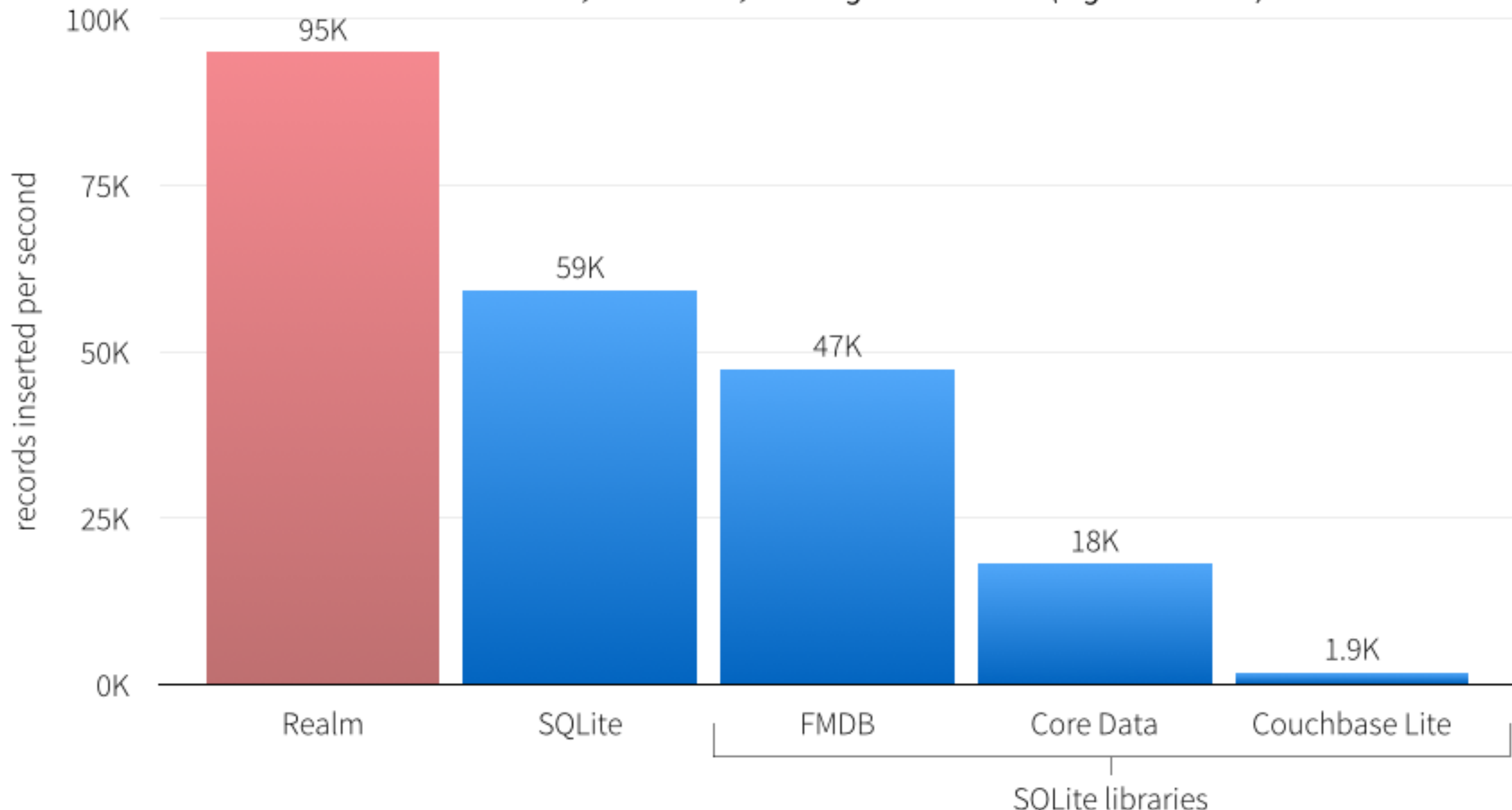
Queries

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 *FTJacks = [Employee objectsWhere:@"fullTime == YES && name == Jack"];
```

Using Realm (Swift)

```
let company = Company() // Using Realm Objects  
company.name = "Realm" // etc...
```

```
defaultRealm().write { // Transactions  
    defaultRealm().add(company)  
}
```

```
// Queries
```

```
let companies = objects(Company)  
companies[0].name // => Realm (generics)  
let ftJacks = objects(Employee) // "Jack"s who work full time  
    .filter(.fullTime == true && .name == "Jack")
```

Work In Progress

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

Questions?

@simjp, jp@realm.io