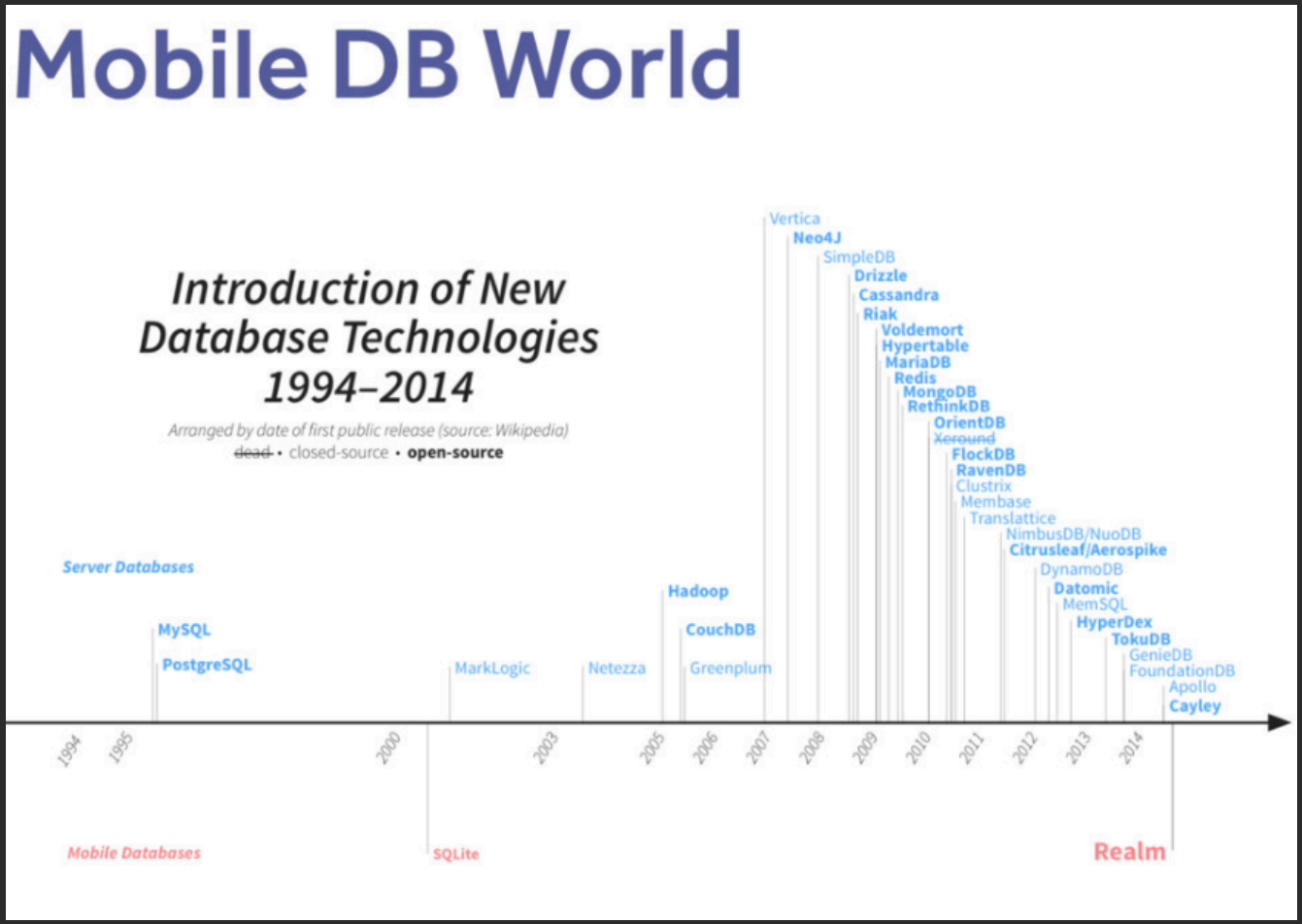
Realm

Realm [relm]

- 렐름 (X) 리얼엠 (X) 렘 (O)
- C++ 베이스
- 2014년 7월 15일 첫 공개 (현재 최신 버전 2.8.3)
- 1. Fast , **2. Easy to use** , 3. Features
- 오픈소스(무료!) + 한글화 + 커뮤니티

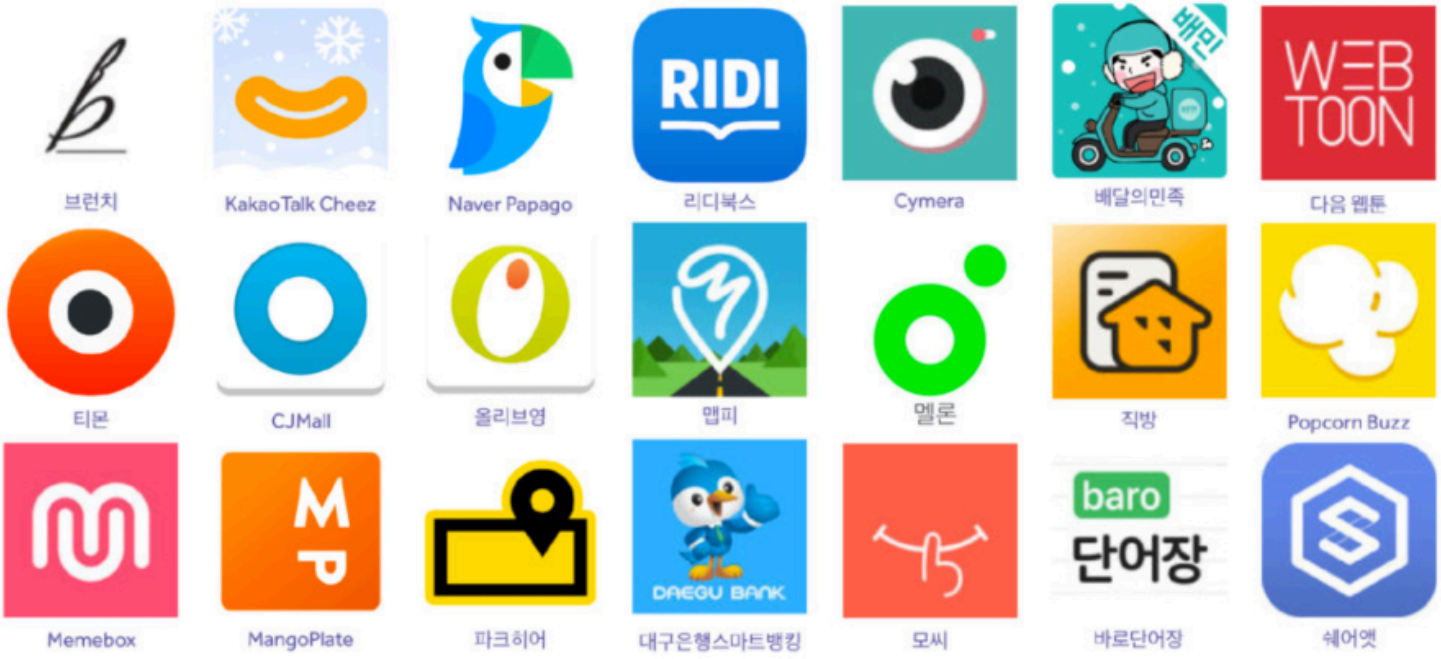
Mobile DB World



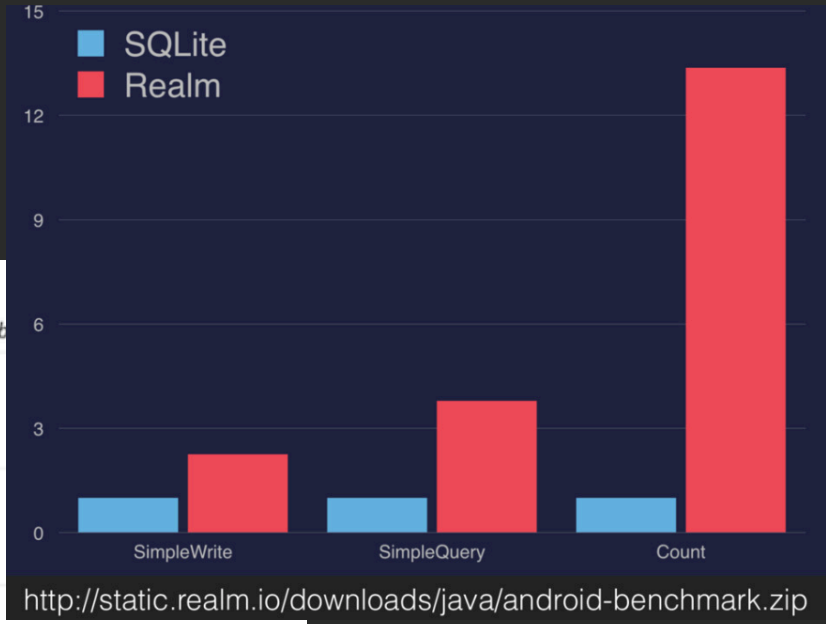
2014년 7월 15일 첫 공개


amazon
Google
hipmunk
STARBUCKS
ebay
REI co-op
NETFLIX
intel
HYATT
BBC
GoPro
SAP
L'ORÉAL
adidas
Alibaba
AVIS
Dropbox
cisco
Walmart
Virgin
NIKKEI
IBM
LINE
intuit
Budweiser
zynga
SoftBank
SIEMENS
RITE AID
SONY
zipcar
AstraZeneca
NHK
imgur
abc
Meetup
AP
UNDER ARMOUR
macy's
GM
TARGET

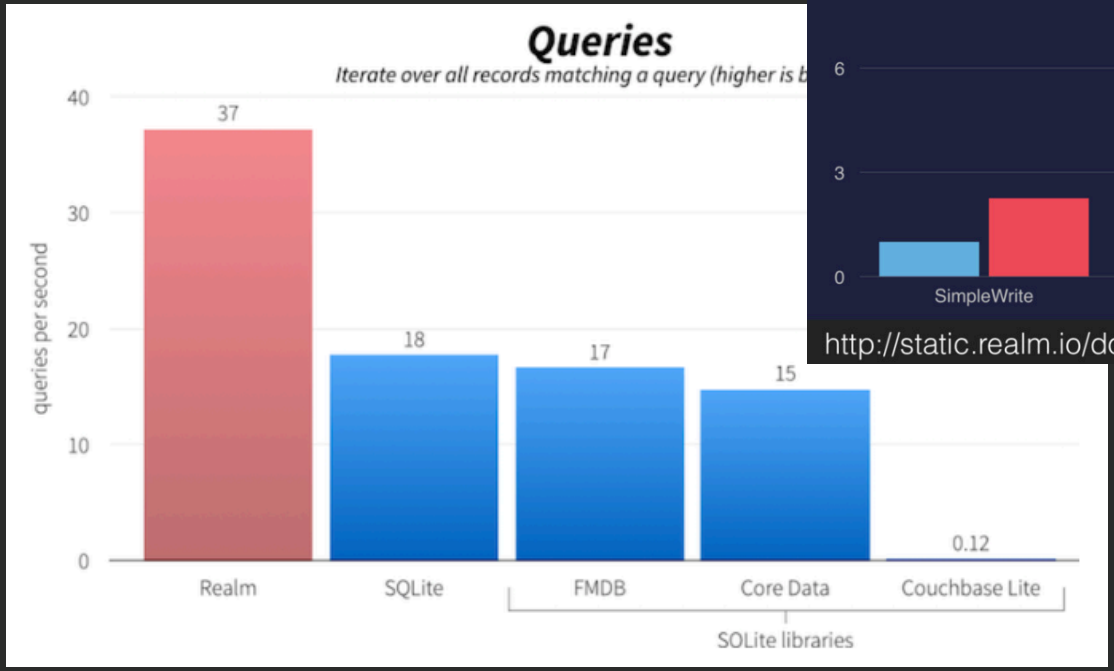
Korean iOS Apps using Realm



Realm 성능 벤치마크



http://static.realm.io/downloads/java/android-benchmark.zip



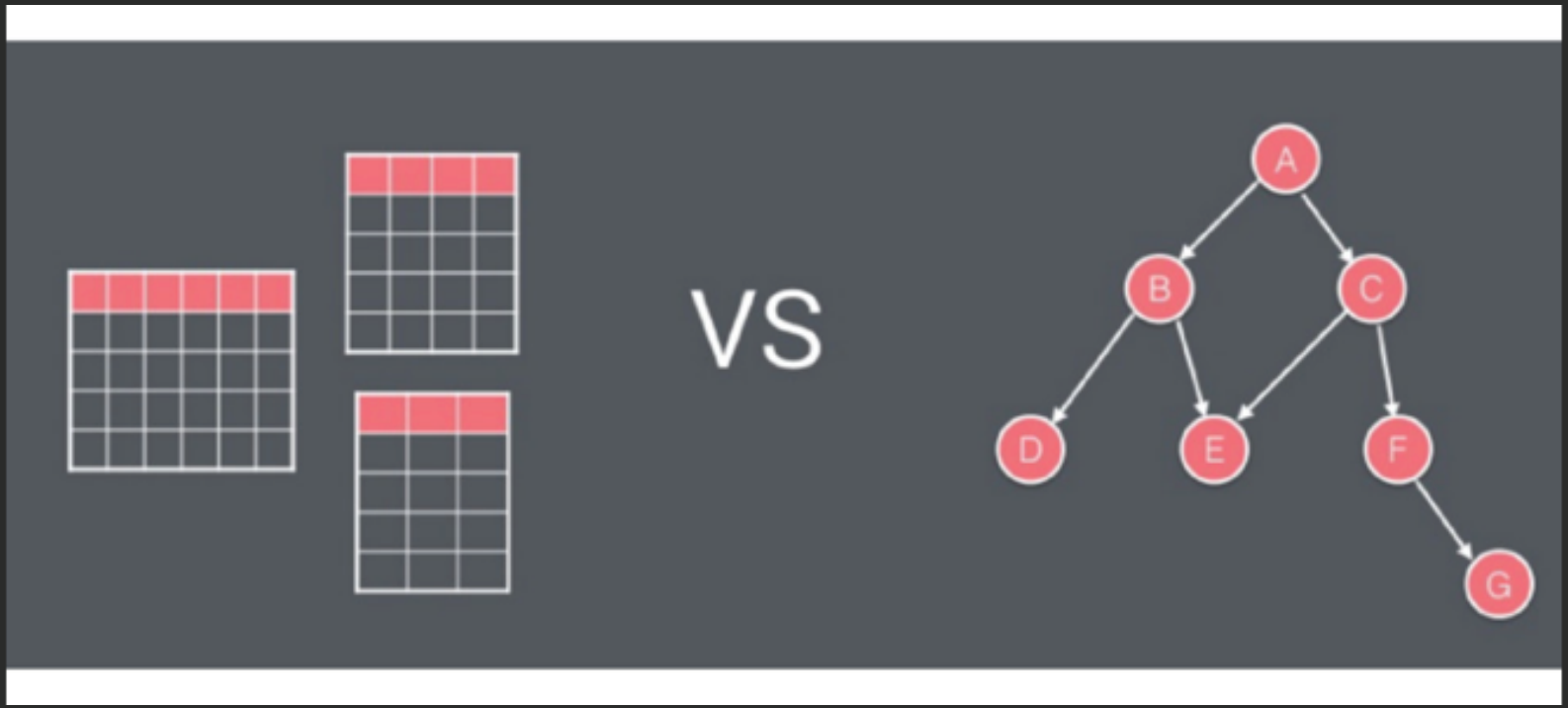
Realm 특징

- 매우 큰 B+ 트리를 **컬럼!** 기준으로 구성
- MVCC (Multiversion Concurrency Control) 타입 DB
  - 항상 마지막 버전만 사용
- 라이브 객체 - Key-Value Store (X), ORM (X), 데이터 저장/로드 시 변환X. 인스턴스 그대로 작업. 항상 동기화
- 변경 발생 - 쓰기 시점에 복사(copy-on-write)
- 앱 런칭 시 모델 유효성 검사. 사용하지 않아도 모든 필드는 유효해야 하며 Non-Optional 타입은 값 필수. 인스턴스는 사용 시점에 로드됨.
- Realm 파일 용량 제한 없음. Property 하나의 최대 크기는 16MB

컬럼 기준 저장 및 조회

인덱스	이름	이메일	전화번호
1	디카프리오	d@realm.io	1004
2	임은주	da@realm.io	1005
3	게이츠	g@realm.io	1006
4	갓민우	1@realm.io	1007

Object Store

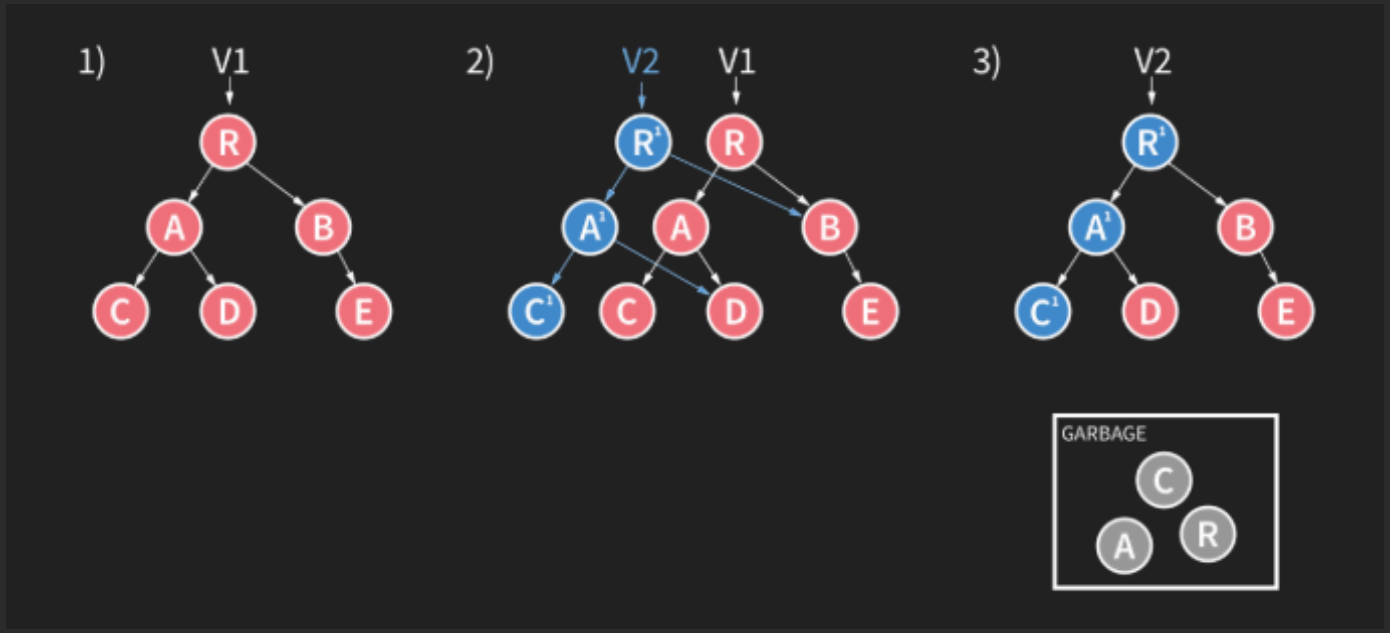


RDBMS

MVCC

MVCC Transaction

변경 발생 -> 쓰기 시점 복사 (copy-on-write)
트리를 포크하여 기존 데이터 수정 없이 복사



MVCC Transaction

스레드 로컬 1: 버전 1

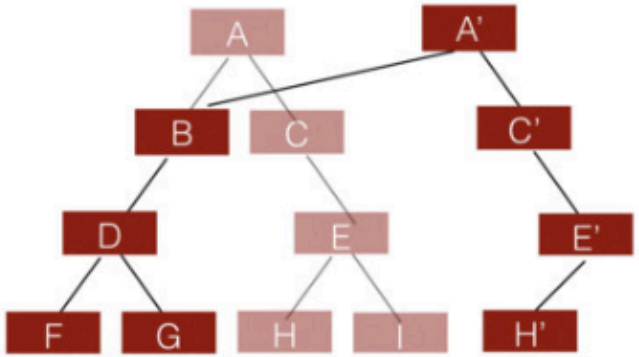
유저 1: 버전 1 읽기 중

스레드 로컬 2: 버전 1

유저 2: 버전 1 읽기 중

유저 3:버전 2 쓰기중

스레드 로컬 3: 버전 1 -> 2

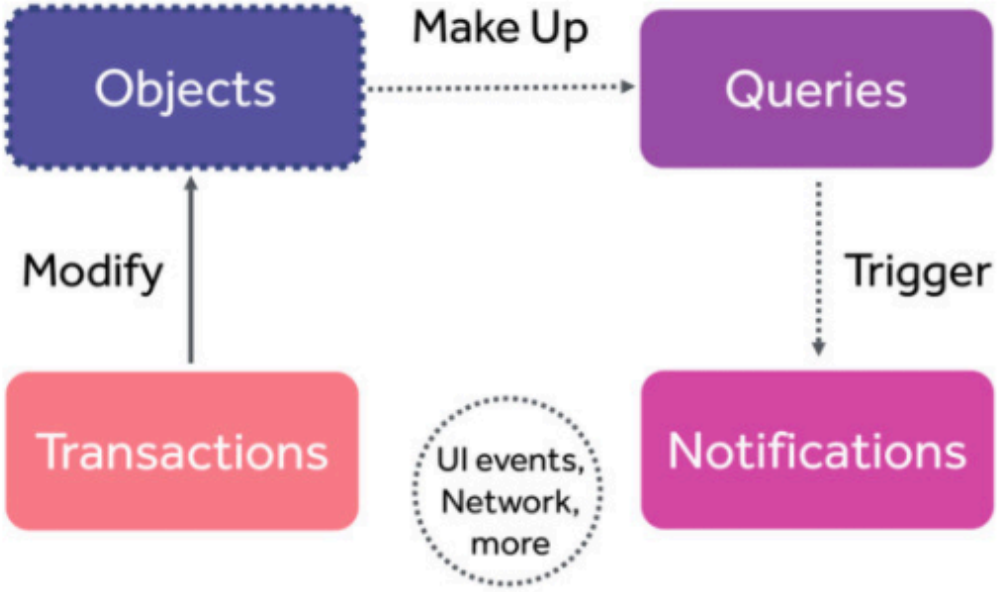


혼란스러울 수 있는 스레드 로컬

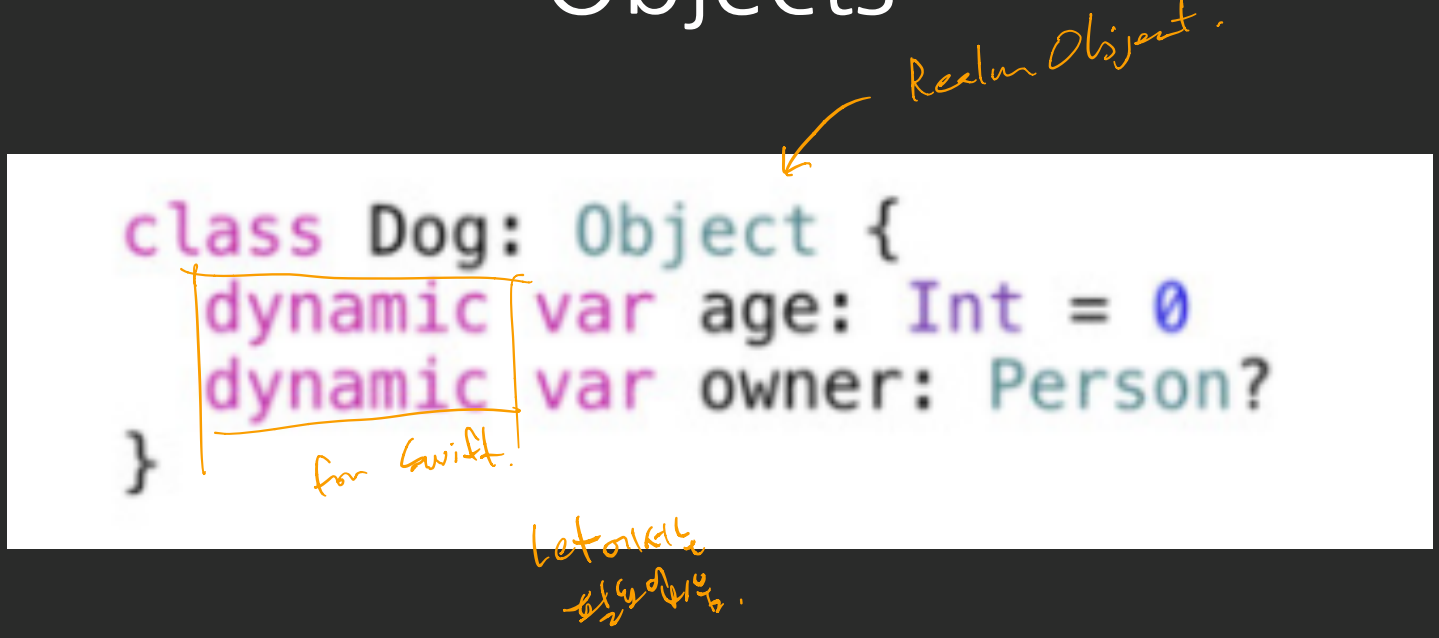
- 배타적이지 않게 효율적으로 읽기, 쓰기를 하기 위해
  메타 데이터를 스레드 단위로 저장.
- 메모리 맵으로 연결된 데이터는 멀티 스레드로 공유되고 있음.
- 다만 스레드 로컬로 열린 Realm, RealmList, RealmResults, RealmObject
  객체를 다른 스레드로 전달할 수 없음.
- 다른 스레드에서 쿼리를 하면 멀티 스레드 자료 공유 덕에 바로 읽음.

Realm 기본 API

Realm Mobile Database



Objects



- 인스턴스 자체가 변환과정 없이 DB 에 저장 (제로카피)

Queries

```
let puppies = realm
  .objects(Dog.self)
  .filter("age < 2")
```

- 실제 작업이 일어나는 것이 아니라 Query 선언 단계 (let 참고)
- 질의는 실제 사용 시 수행. 따라서 쿼리를 새로 작성하지 않고 그대로 사용해도 결과는 항상 최신 상태 유지 (Live Object)

Notifications

```
let token = puppies.addNotificationBlock { changes in
  self.updateUI(with: changes)
}
```

- Object 에 변경이 발생하면 Notification 발생

Transaction

```
try realm.write {
  dog.owner = Person(name: "me")
}
```

- write 메서드를 통해 트랜잭션 수행
- ACID 보장

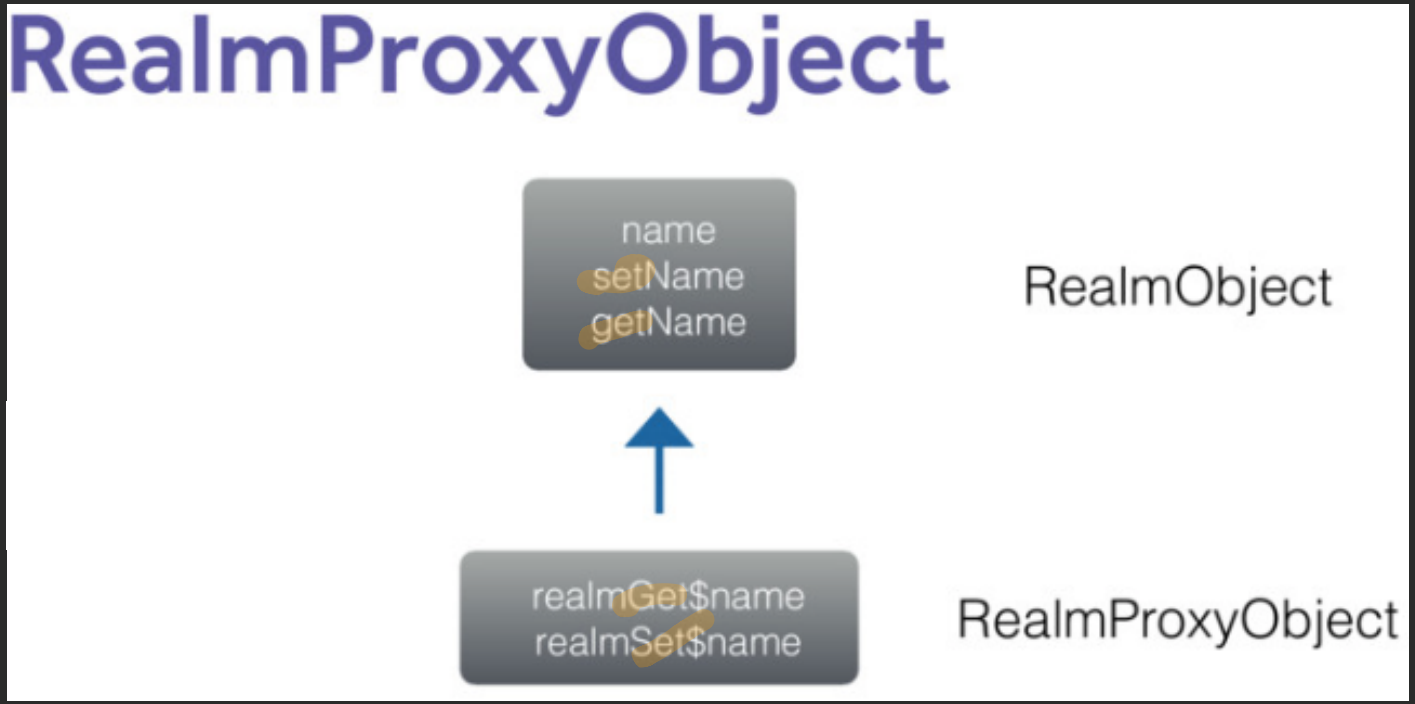
Example

```
// Define your models like regular Swift classes
class Dog: Object {
  dynamic var name = ""
  dynamic var age = 0
}
class Person: Object {
  dynamic var name = ""
  dynamic var picture: NSData? = nil // optionals supported
  let dogs = List<Dog>()
}

// Use them like regular Swift objects
let myDog = Dog()
myDog.name = "Rex"
myDog.age = 1
print("name of dog: \(myDog.name)")
```

Example

```swift
// Get the default Realm
let realm = try! Realm()

// Query Realm for all dogs less than 2 years old
let puppies = realm.objects(Dog.self).filter("age < 2")
puppies.count // => 0 because no dogs have been added to the Realm yet

// Persist your data easily
try! realm.write {
  realm.add(myDog)
}

// Queries are updated in realtime
puppies.count // => 1

// Query and update from any thread
DispatchQueue(label: "background").async {
  autoreleasepool {
    let realm = try! Realm()
    let theDog = realm.objects(Dog.self).filter("age == 1").first
    try! realm.write {
      theDog!.age = 3
    }
  }
}
```

실제 객체



- 굳이 알고 있을 필요는 없어요.

Realm 소개 시리즈

1. Realm DB 엔진의 장점 - https://goo.gl/jcfnWw
2. Realm API 소개 - https://goo.gl/M37dcA
3. 플랫폼 간 코드 공유 - https://goo.gl/uCtCj7
4. 유연하고 견고한 SDK의 장점 -https://goo.gl/rdn31v
5. 노티피케이션 - https://goo.gl/BRgG9U
6. Realm을 통한 모바일 앱 개발 전략 - https://goo.gl/v5Xk4y

추가 참고 링크

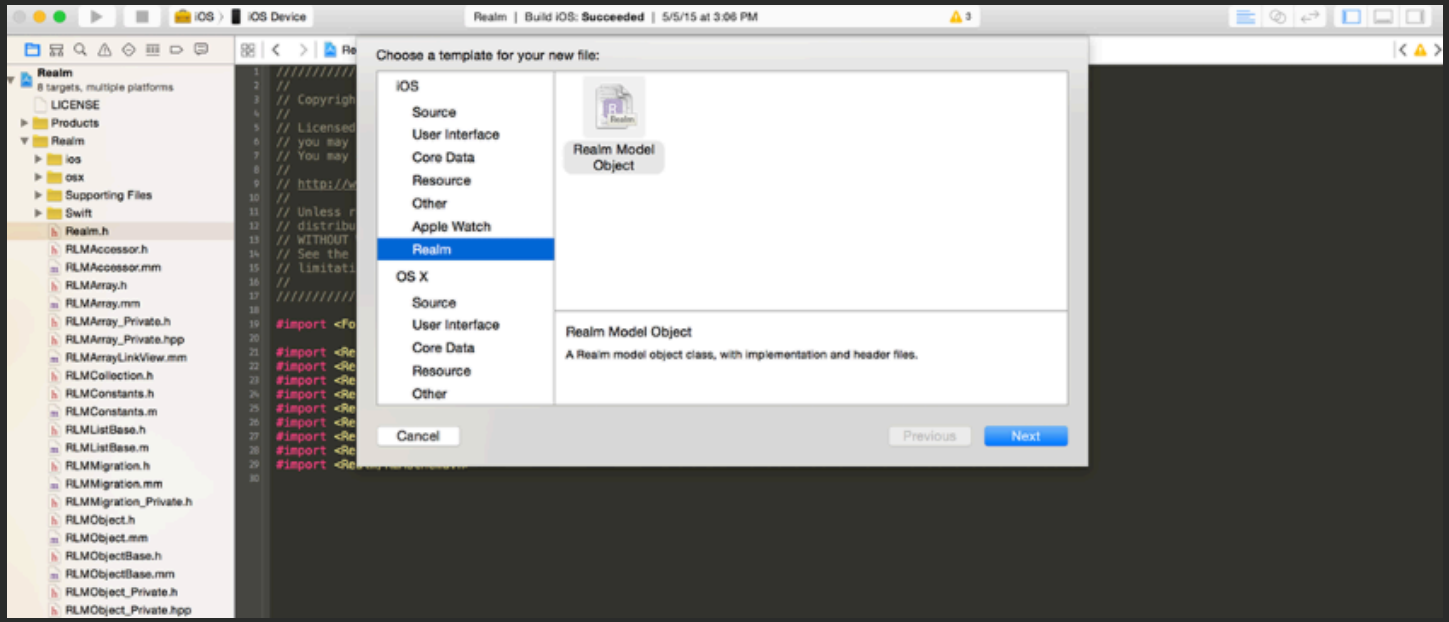
- 왜 SQLite 에서 Realm 으로 옮겼는가? (https://goo.gl/VBCfSP)
- Realm, 제대로 알고 안드로이드에서 사용하기 (http://wp.me/p7cE56-7O)

Realm 관련 링크

1. 홈페이지: realm.io/kr
2. Realm 뉴스: realm.io/kr/news/
3. GitHub: github.com/realm
4. 페이스북 페이지: facebook.com/realmkr
5. 페이스북 그룹: facebook.com/groups/realmkr

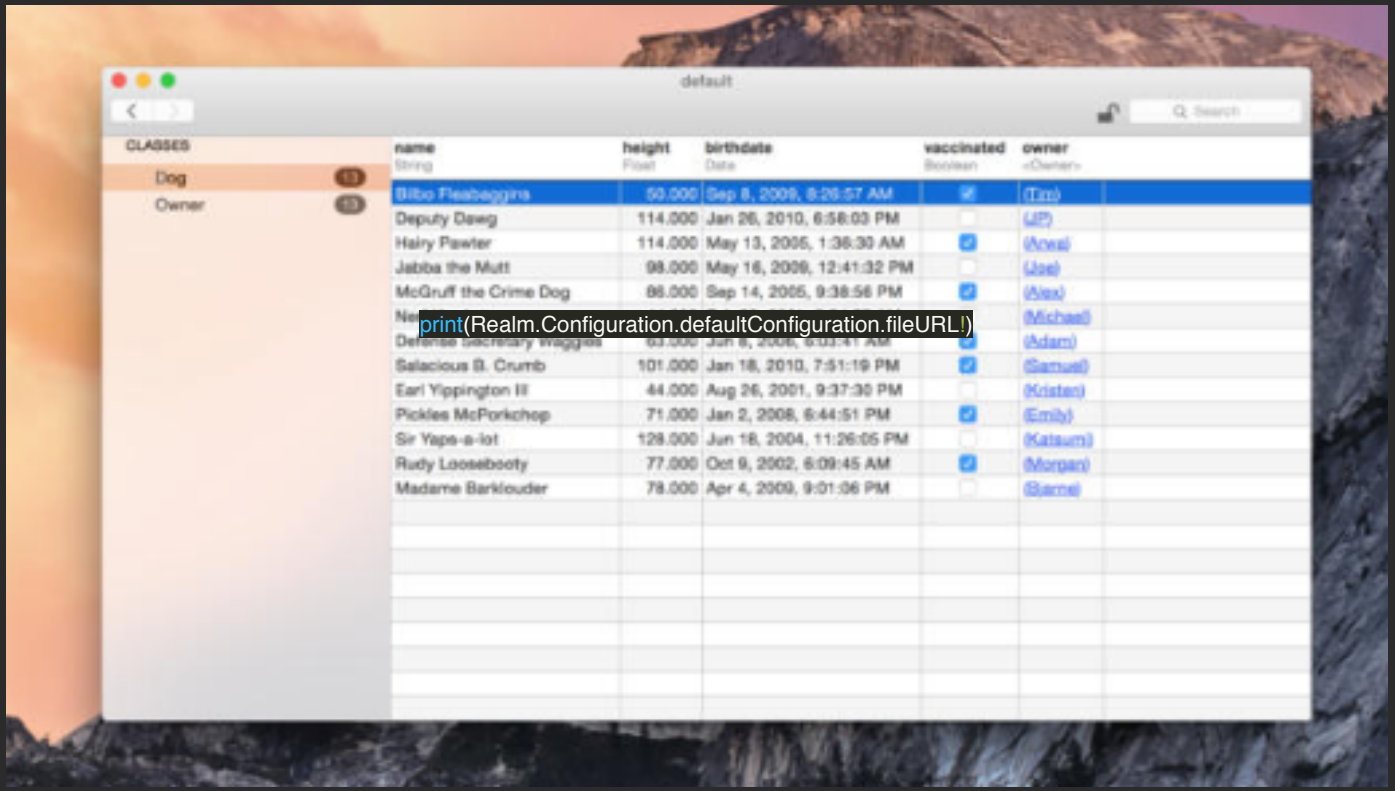
Realm 실습

Realm File Template



1. https://goo.gl/6RFDm7
2. release.zip -> plugin/RealmPlugin.xcodeproj 실행
3. 빌드 후 Xcode 재실행

Realm Browser



https://goo.gl/xZSjVh

Realm 파일 경로

```
print(Realm.Configuration.defaultConfiguration.fileURL!)
```

참고 링크

Realm 파일 경로 찾기 - https://goo.gl/Vps1pe

Migration

```
Realm.Configuration.defaultConfiguration = Realm.Configuration(
  schemaVersion: 2,
  migrationBlock: { migration, oldSchemaVersion in
    // The enumerateObjects:block: method iterates
    // over every 'Person' object stored in the Realm file
    migration.enumerateObjects(ofType: Person.className()) { oldObject, newObject in
      // Add the `fullName` property only to Realms with a schema version of 0
      if oldSchemaVersion < 1 {
        let firstName = oldObject!["firstName"] as! String
        let lastName = oldObject!["lastName"] as! String
        newObject!["fullName"] = "\(firstName) \(lastName)"
      }

      // Add the `email` property to Realms with a schema version of 0 or 1
      if oldSchemaVersion < 2 {
          newObject!["email"] = ""
      }
    }
  })
```

참고 링크

- Migration 공식 예제 - http://bit.ly/2tgI0b7
- Realm Migration 설정 - https://goo.gl/SxujzE