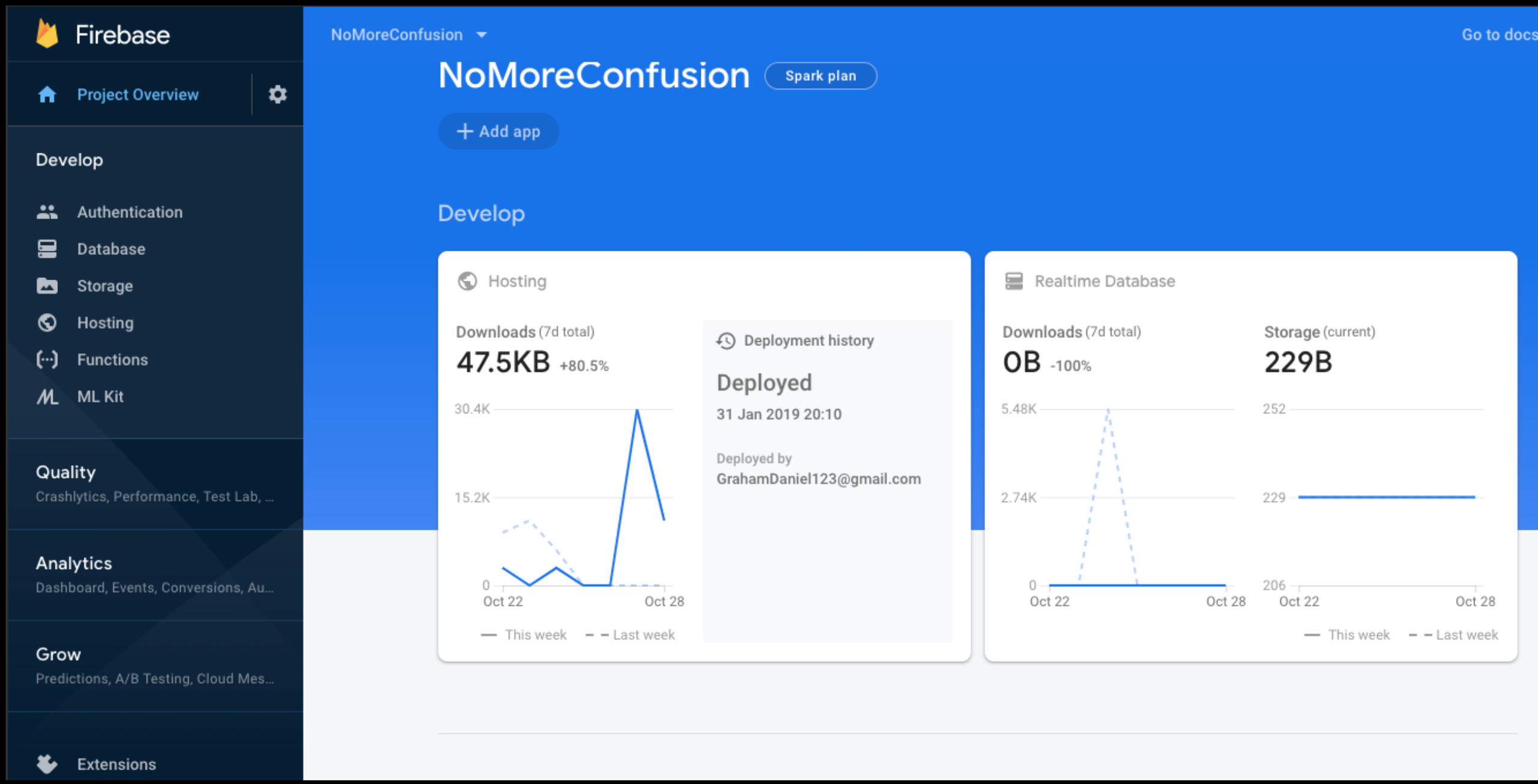
DANIEL GRAHAM

CLOUD FIRESTORE

PART I

WHAT IS FIRESTORE/FIREBASE

NO SQL DATABASE DESIGN TO RUN IN CLOUD



WHAT IS A DATABASE

- “a structured set of data held in a computer, especially one that is accessible in various ways.”

USER

KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019

- Data is stored in tables columns and rows

USER

COLUMN

KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019

ROW

DATA

- The structure of each table is fixed when it is created
- Each new row of data must contain same number of columns
- The data type of the columns must match

USER SCHEMA

KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019
3	2	2/2/2019

ALL VALUES IN THE COLUMN MOST BE SAME TYPE

BECAUSE THE THESE RESTRICTIONS

- Because the schema restriction normally table as design so store objects of the type for example

KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019

KEY	REVIEW	COMMENT
1	4	“YUMMY”
2	1	“YUCKY”

BECAUSE THE THESE RESTRICTIONS

- Because the schema restriction normally table as design so store objects of the type for example

KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019

KEY	REVIEW	COMMENT
1	4	“YUMMY”
2	1	“YUCKY”

BECAUSE THE THESE RESTRICTIONS

- Because the schema restriction normally table as design so store objects of the type for example

KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019

FOREIGN KEY

KEY	REVIEW	COMMENT	ID
1	4	“YUMMY”	1
2	1	“YUCKY”	1

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

DINER TABE

REVIEW TABLE

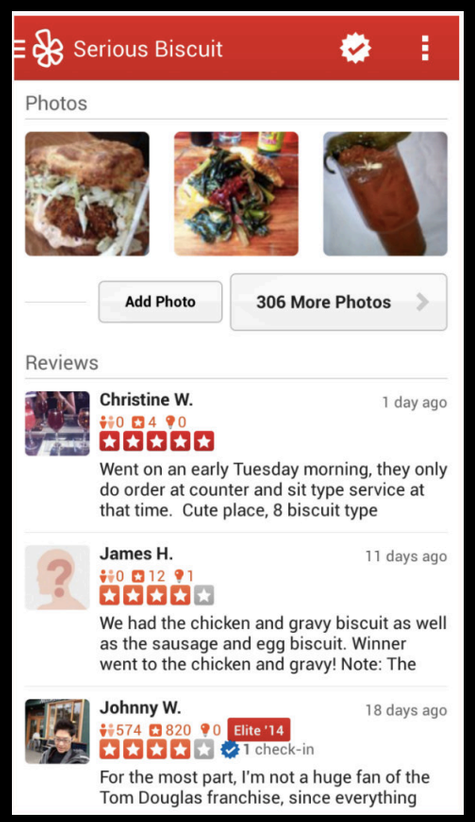
KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

USER TABLE

Relational Diagram

SO HOW WOULD I SHOW MY RESTAURANT INFORMATION SCREEN WITH ALL THE USERS AND REVIEW DATA



DINER TABE

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

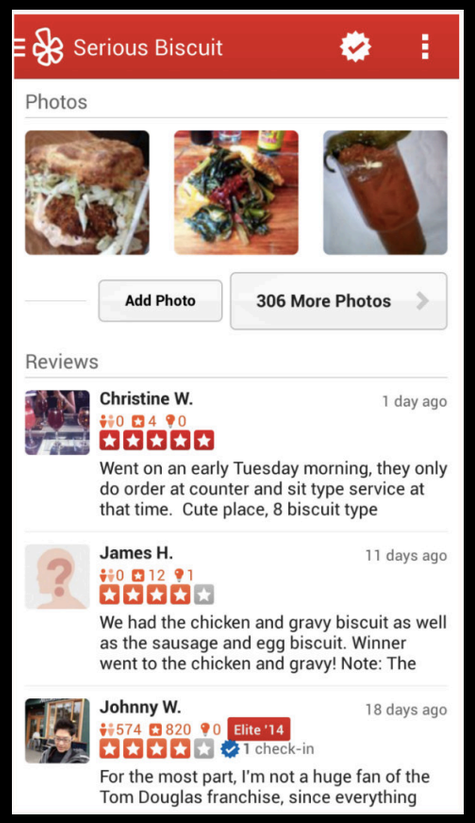
REVIEW TABLE

KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

USER TABLE

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

SO HOW WOULD I SHOW MY RESTAURANT INFORMATION SCREEN WITH ALL THE USERS AND REVIEW DATA



DINER TABE

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

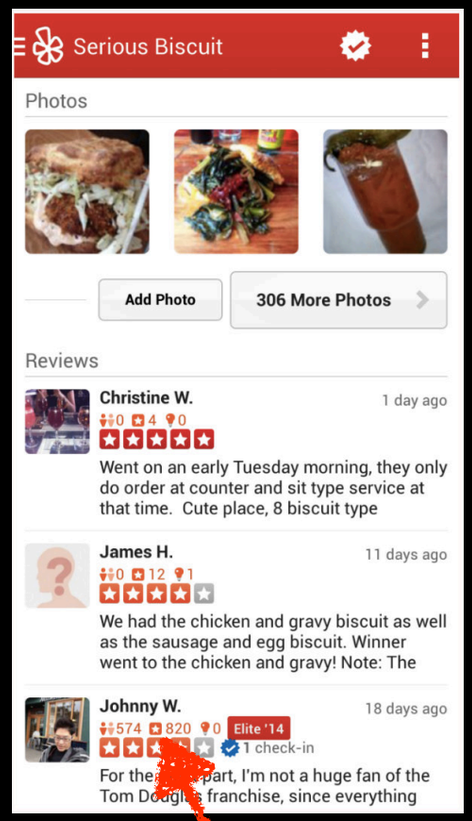
REVIEW TABLE

KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

USER TABLE

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

SO HOW WOULD I SHOW MY RESTAURANT INFORMATION SCREEN WITH ALL THE USERS AND REVIEW DATA



MIGHT
WHAT TO SHOW USER
INFO SO INCLUDE

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

SO HOW WOULD I SHOW MY RESTAURANT INFORMATION SCREEN WITH ALL THE USERS AND REVIEW DATA

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

KEY	REVIEW	COMMEN	UID	R D
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

1	"OHILL"	NORTH

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

SO HOW WOULD I SHOW MY RESTAURANT INFORMATION SCREEN WITH ALL THE USERS AND REVIEW DATA

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

KEY	REVIEW	COMMEN	UID	R	D
1	4	"YUMMY	1	1	
2	1	"YUCKY"	1	2	

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

1	"OHILL"	NORTH

1	4	"YUMMY	1	1

SO HOW WOULD I SHOW MY RESTAURANT INFORMATION SCREEN WITH ALL THE USERS AND REVIEW DATA

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

1	"OHILL"	NORTH

1	4	"YUMMY	1	1

1	"BOT99"	2/11/2019

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

SQL

1	"OHILL"	NORTH		
1	4	"YUMMY"	1	1
1	"BOT99"	2/11/2019		

KEY	NAME	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH

KEY	REVIEW	COMMEN	UID	RID
1	4	"YUMMY	1	1
2	1	"YUCKY"	1	2

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

```
SELECT * FROM DINERS,
REVIEWS, USERS
WHERE DINERS.KEY = 1
AND
REVIEWS.RID = DINER.KEY
AND
REVIEWS.UID = USERS.KEY
```

1	"OHILL"	NORTH		
1	4	"YUMMY"	1	1
1	"BOT99"	2/11/2019		

THINGS ARE DIFFERENT IN NO SQL

KEY	DINER	LOCATION
1	"OLD HILL"	NORTH
2	"CAV"	SOUTH

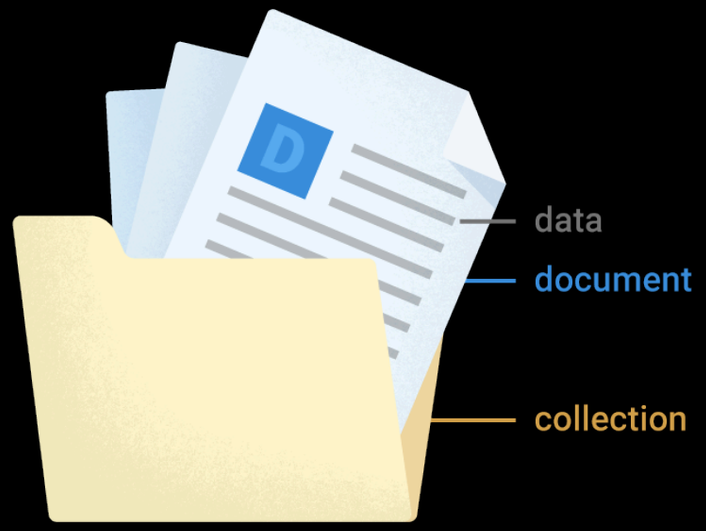
KEY	REVIEW	COMMENT	UID	RID
1	4	"YUMMY"	1	1
2	1	"YUCKY"	1	2

KEY	USERNAME	LOGIN_DATE
1	"BOT99"	2/11/2019
2	"HUMAN2"	10/2/2019

THERE ARE NO TABLES

THINGS ARE DIFFERENT IN NO SQL

KEY	DINER	LOCATION
1	"OHILL"	NORTH
2	"CAV"	SOUTH



{ key: 1
diner: "ohill"
location: "north"
}

{ key: 2
diner: "CAV"
location: "south"
}

DINER COLLECTION

DINER DOCUMENT

Trees Common in real-time databases

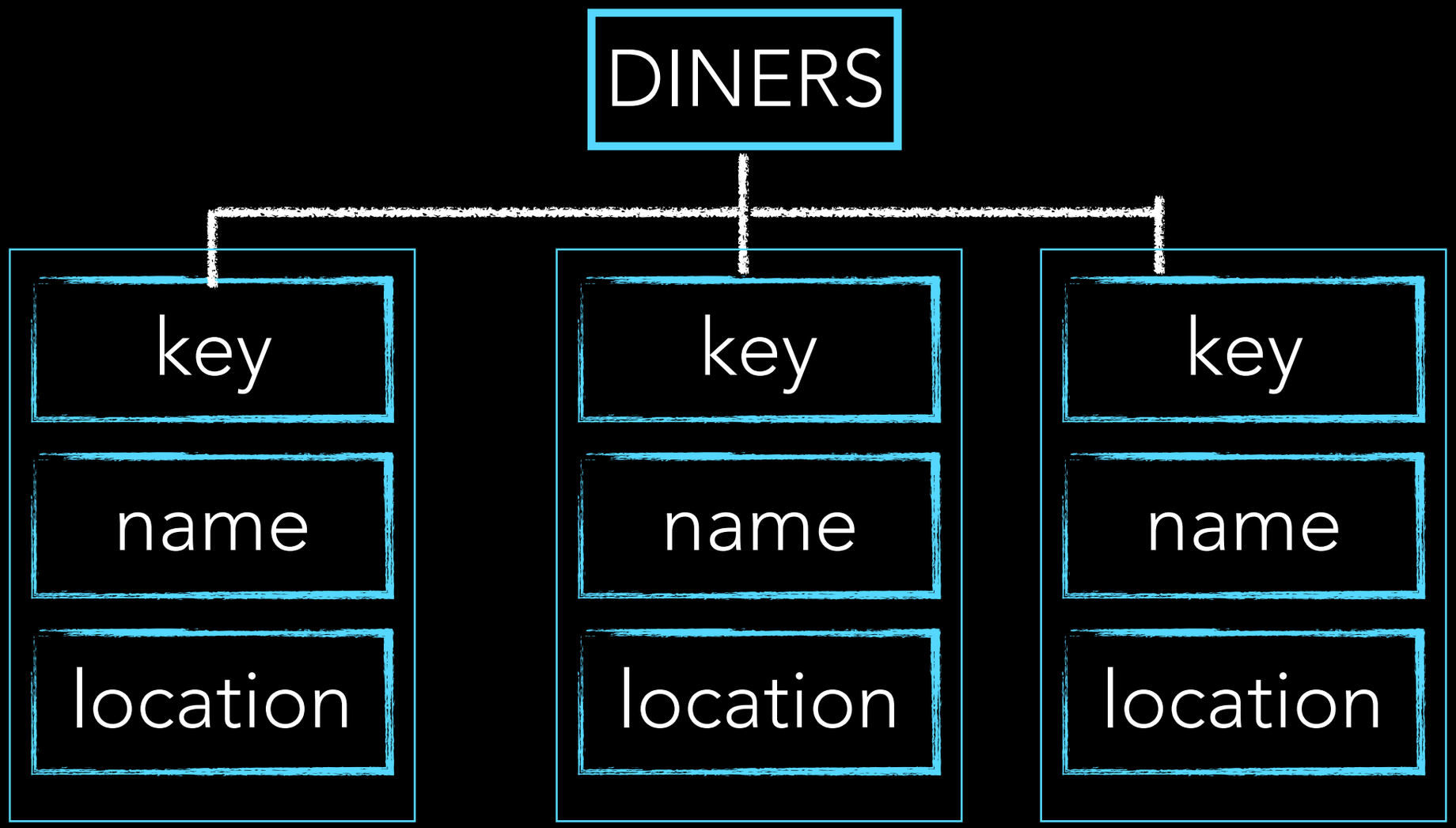
KEY	NAME	LOCATION
1	“OHILL”	NORTH
2	“CAV”	SOUTH

KEY	REVIEW	COMMEN	UID	RID
1	4	“YUMMY	1	1
2	1	“YUCKY”	1	2

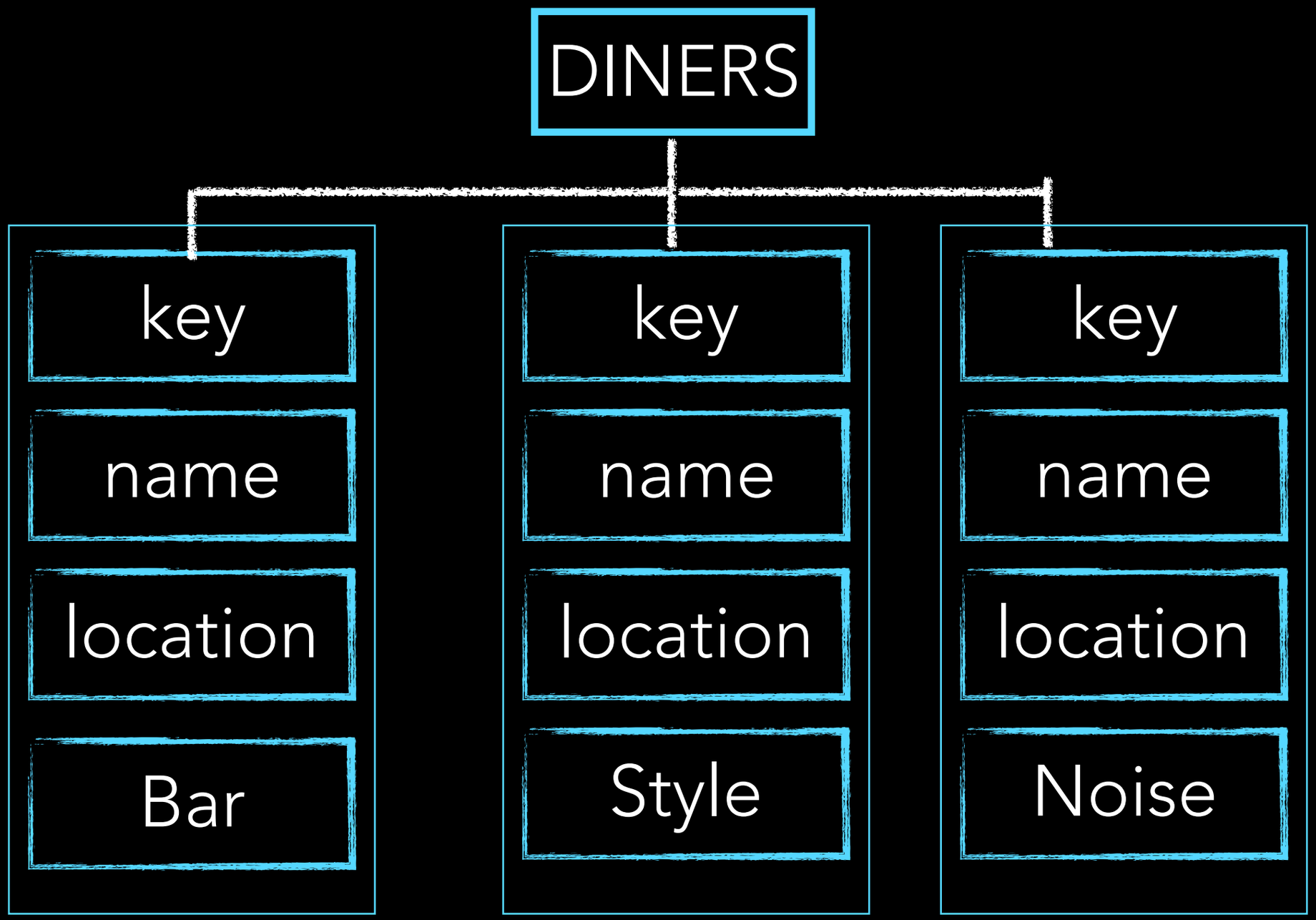
KEY	USERNAME	LOGIN_DATE
1	“BOT99"	2/11/2019
2	“HUMAN2”	10/2/2019

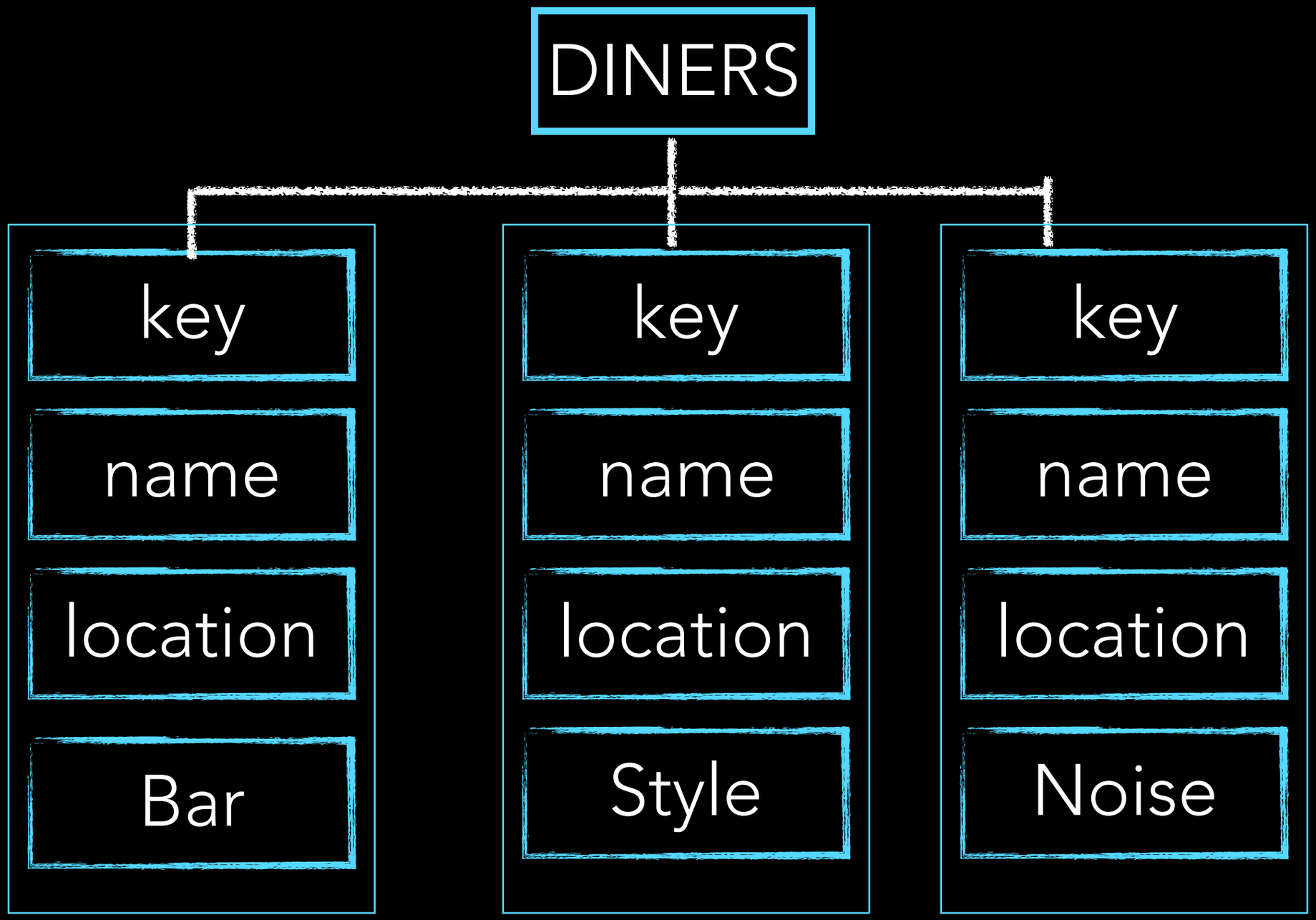
```
{
    "dinners":[
        {
            "key":1,
            “name":"ohill",
            "location":"north",
            "reviews":[
                {
                    "key":1,
                    "review":4,
                    "comment":"yummy",
                    "user":{
                        "key":1,
                        "username":"BOT99",
                        "login_date":"2/11/2019"
                    }
                }
            ]
        }
    ]
}
```

- In NO SQL DATA there are no restriction on what you can place in the database



- In NO SQL DATA there are no restriction on what you can place in the database





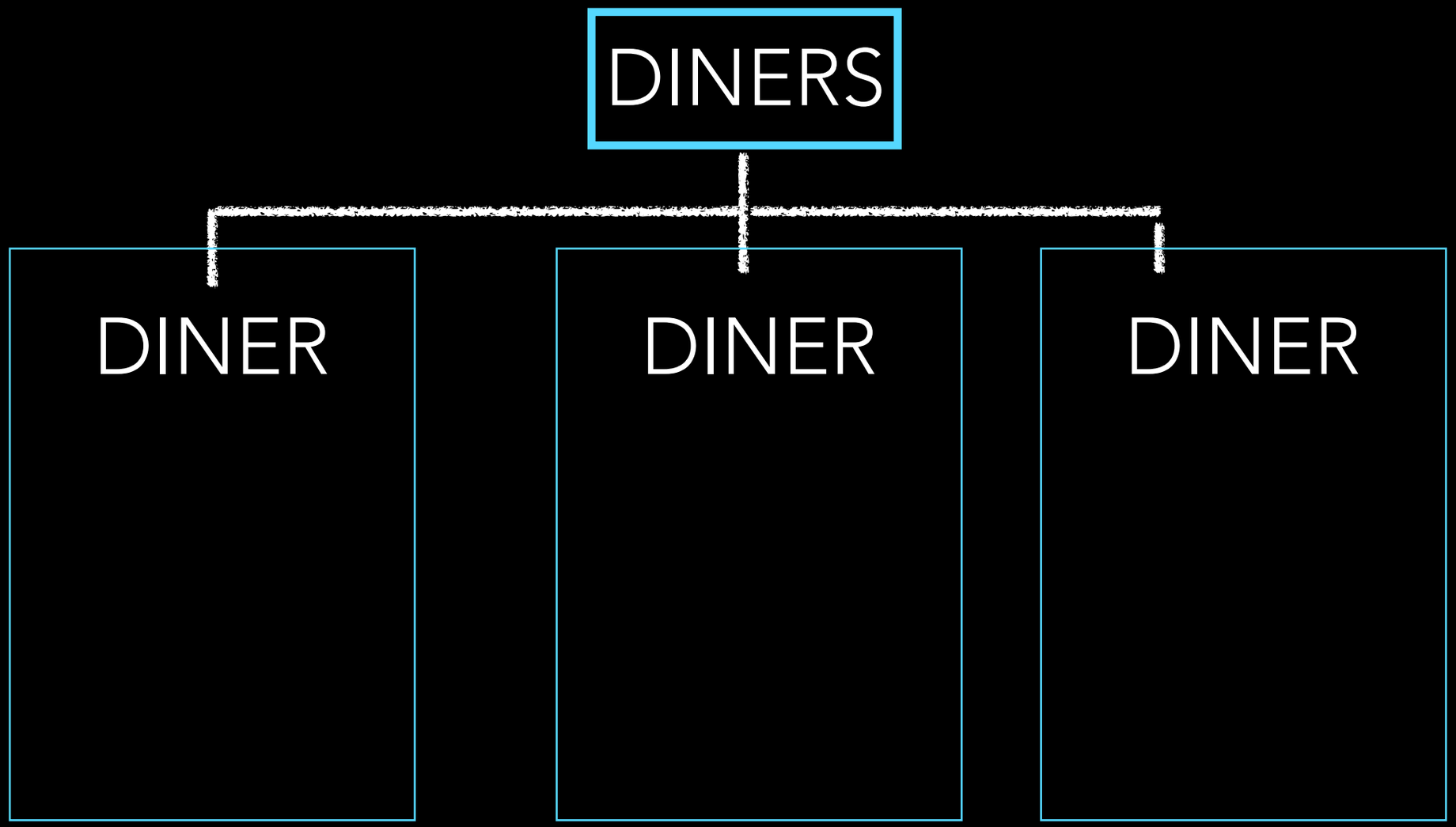
NO Guarantee at the data level of fields contained in your object. So code defensively and check

NO SQL DATABASES DON'T USE SQL

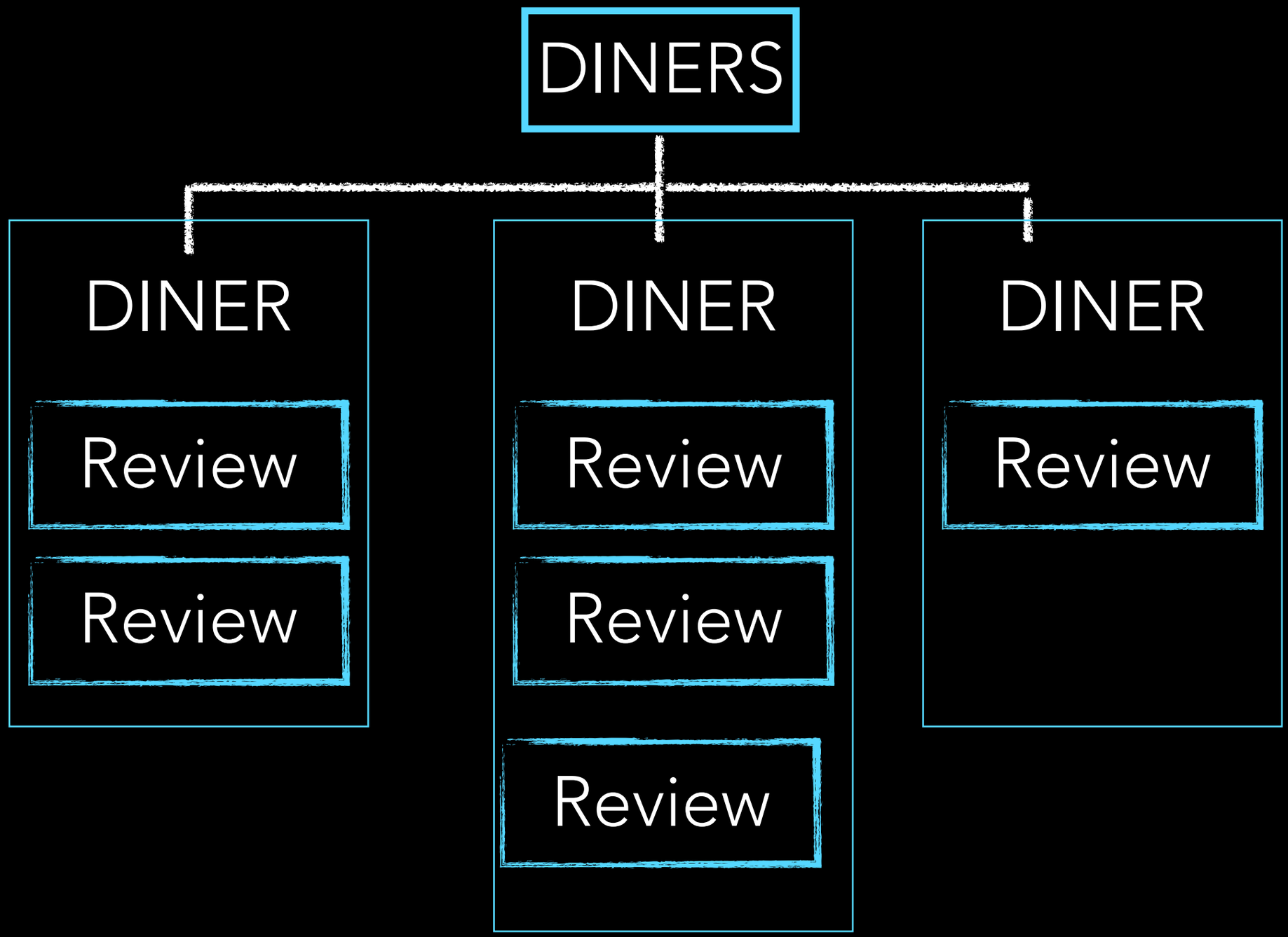
DIFFICULT TO DO JOINS

Typically going to Retrieving a collection

LOOK AT NO SQL VERSION OF DINER REVIEW APP



WILL HAVE DUPLICATED DATA: USERID SO IF A USER CHANGES IDS WILL HAVE MAKE CHANGES ACROSS ALL REVIEW DOCUMENTS



DINERS Collection

DINER
D_ID :1

DINER
D_ID :2

DINER
D_ID :3

Review Collection

D_ID :2
Review

D_ID :1
Review

D_ID :1
Review

D_ID :1
Review

EASY TO FETCH
AN INDIVIDUAL DINER

EASY TO FETCH
OR A GROUP OF DINERS

DINERS Collection

DINER
D_ID :1

DINER
D_ID :2

DINER
D_ID :3

Review Collection

Review
D_ID :2

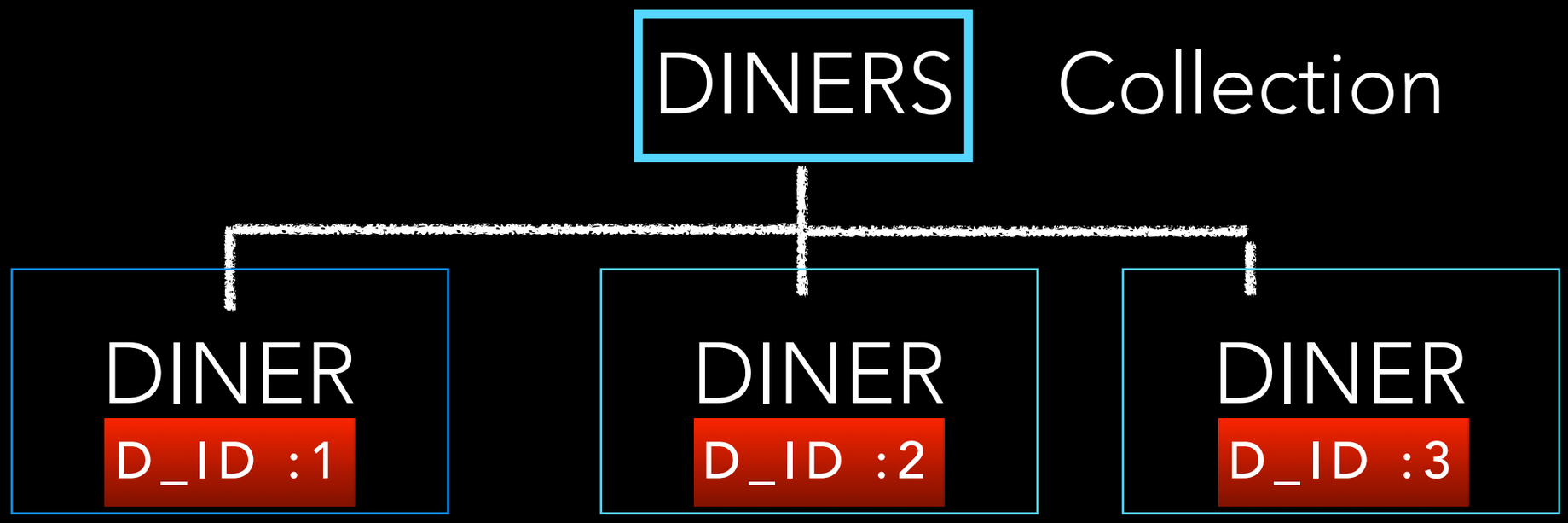
Review
D_ID :1

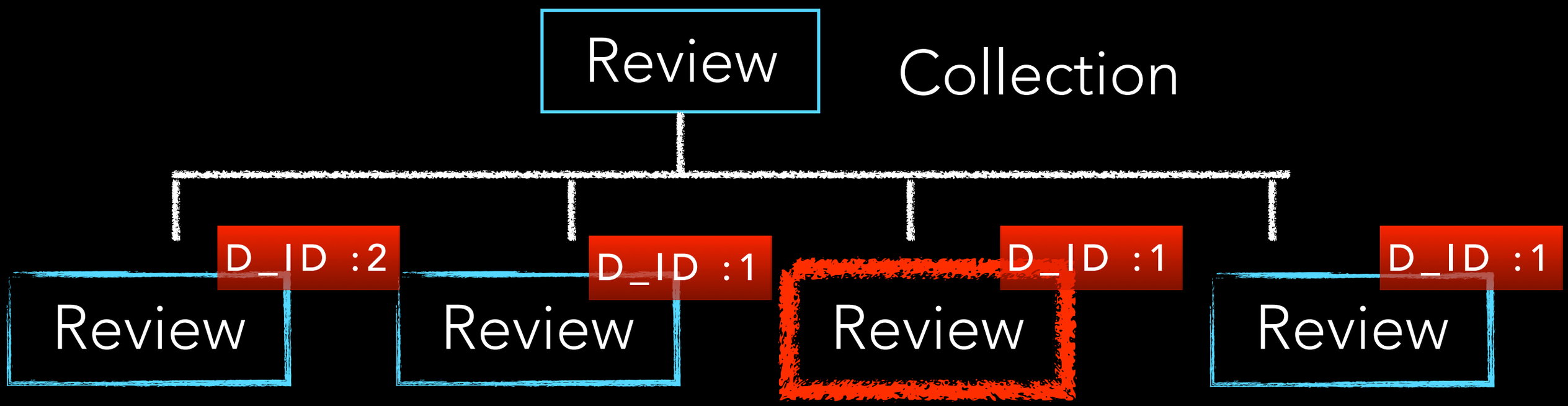
Review
D_ID :1

Review
D_ID :1

Retrieving data

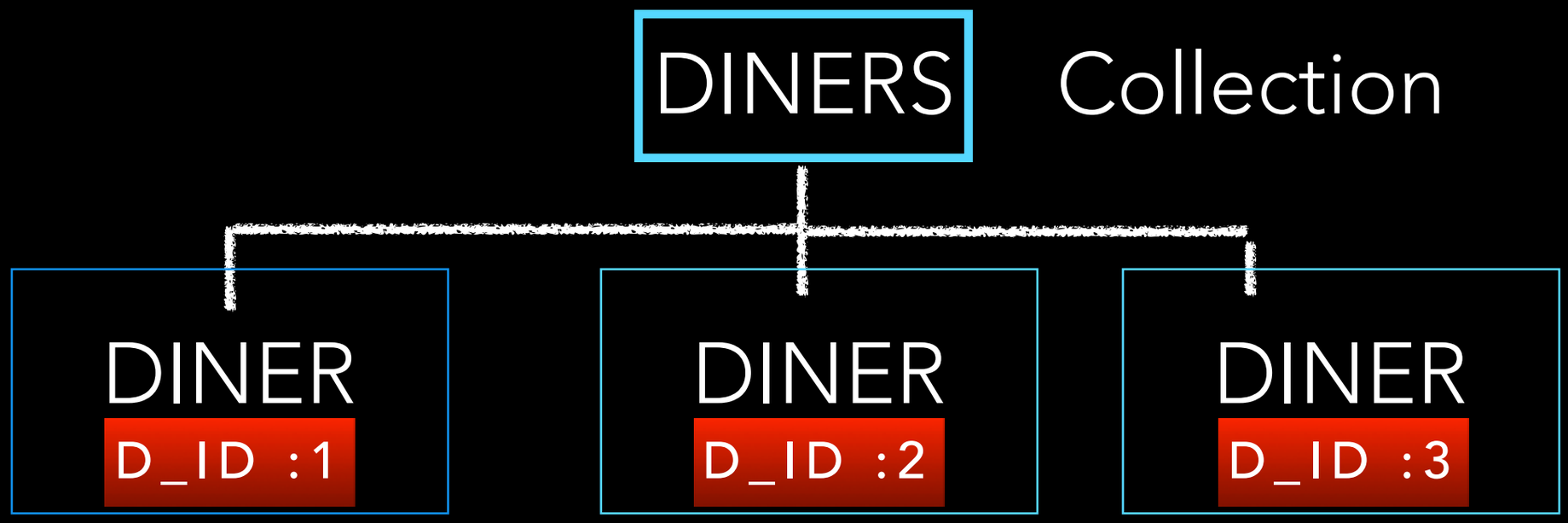
EASY TO FETCH
AN INDIVIDUAL REVIEW

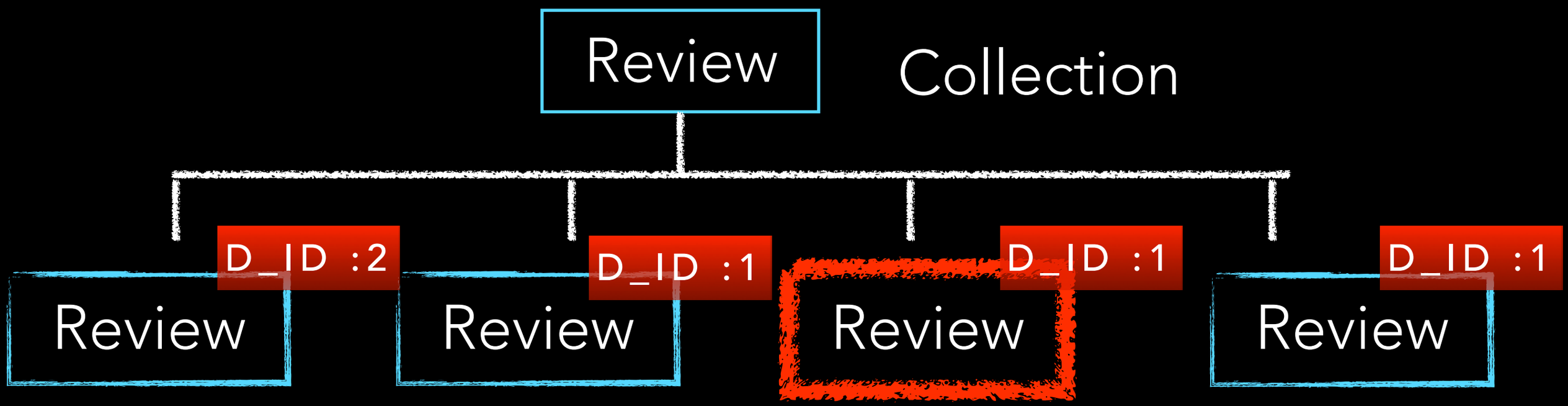




Retrieving data

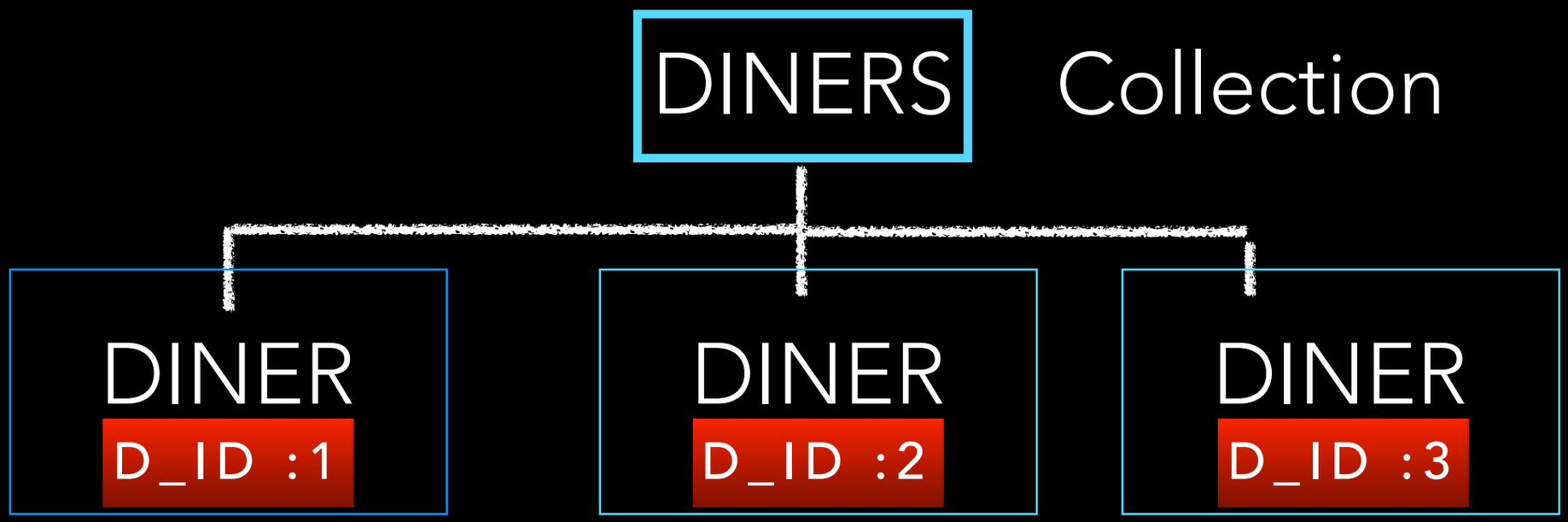
EASY TO FETCH
AN INDIVIDUAL REVIEW

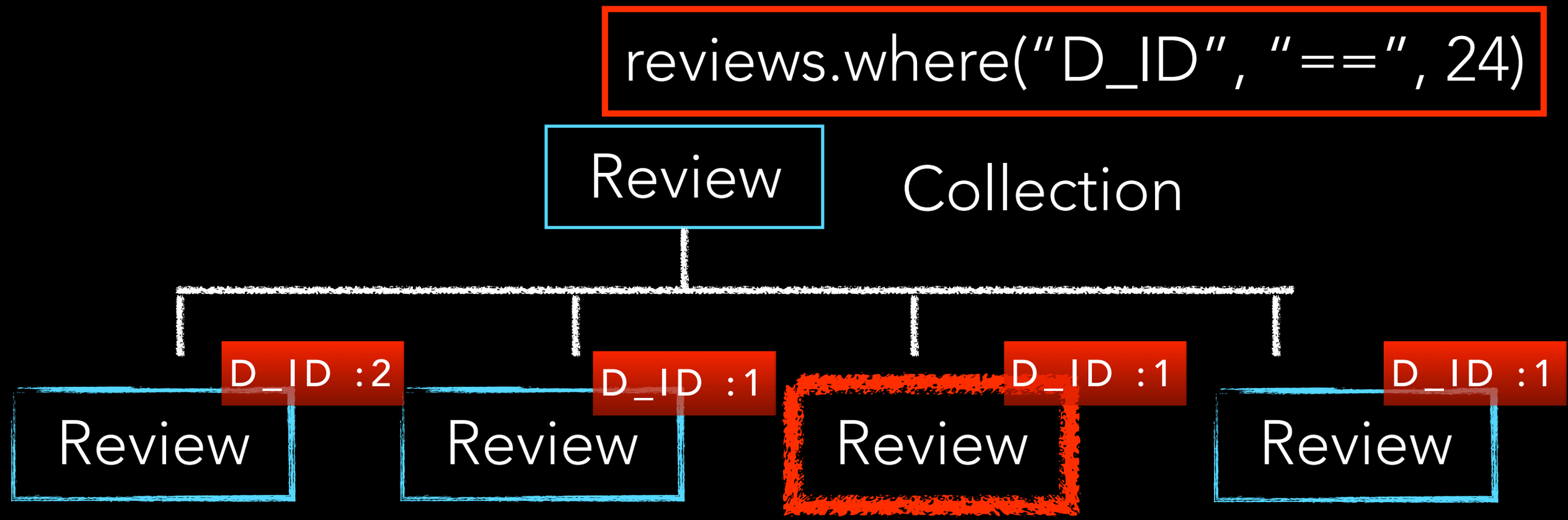




Retrieving data

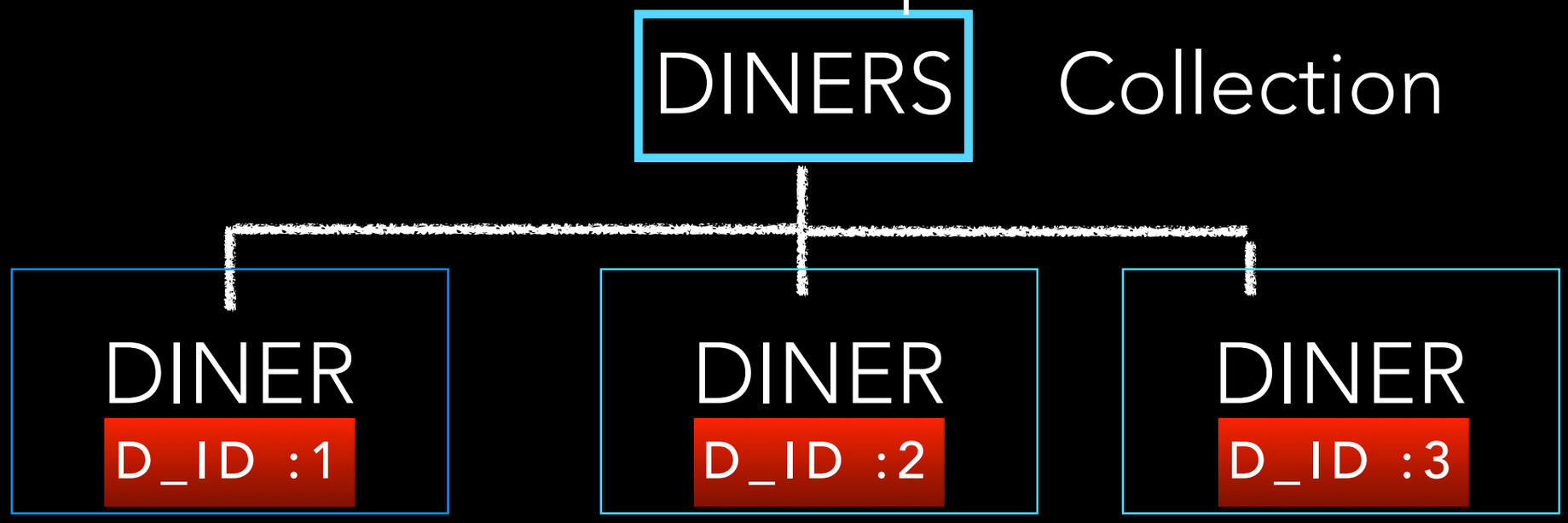
ALSO POSSIBLE TO FETCH ALL REVIEWS FOR AN INDIVIDUAL DINER

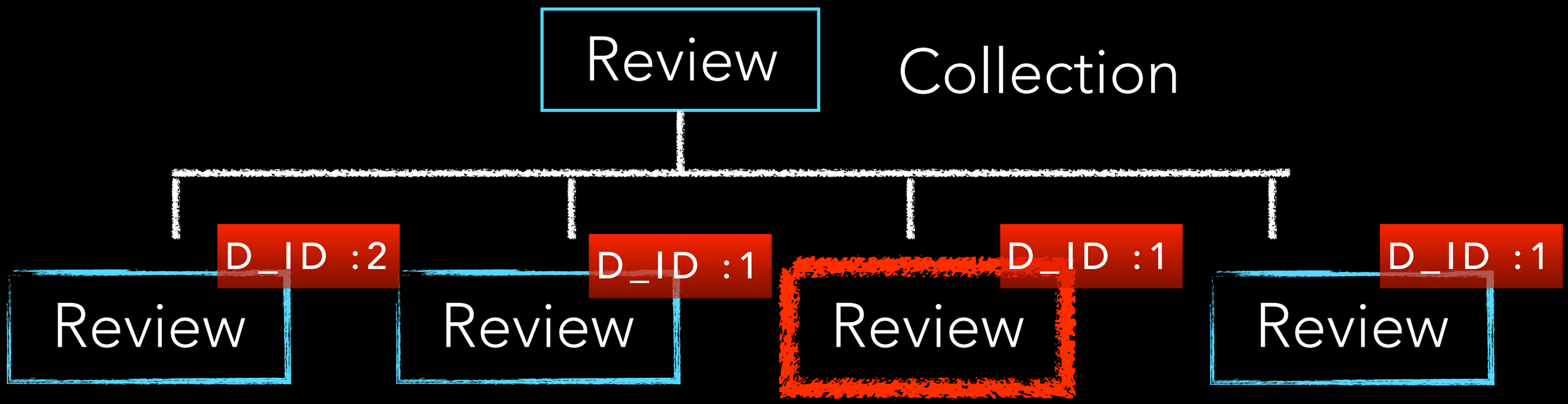




CAN'T request from 2 sub collections need to make

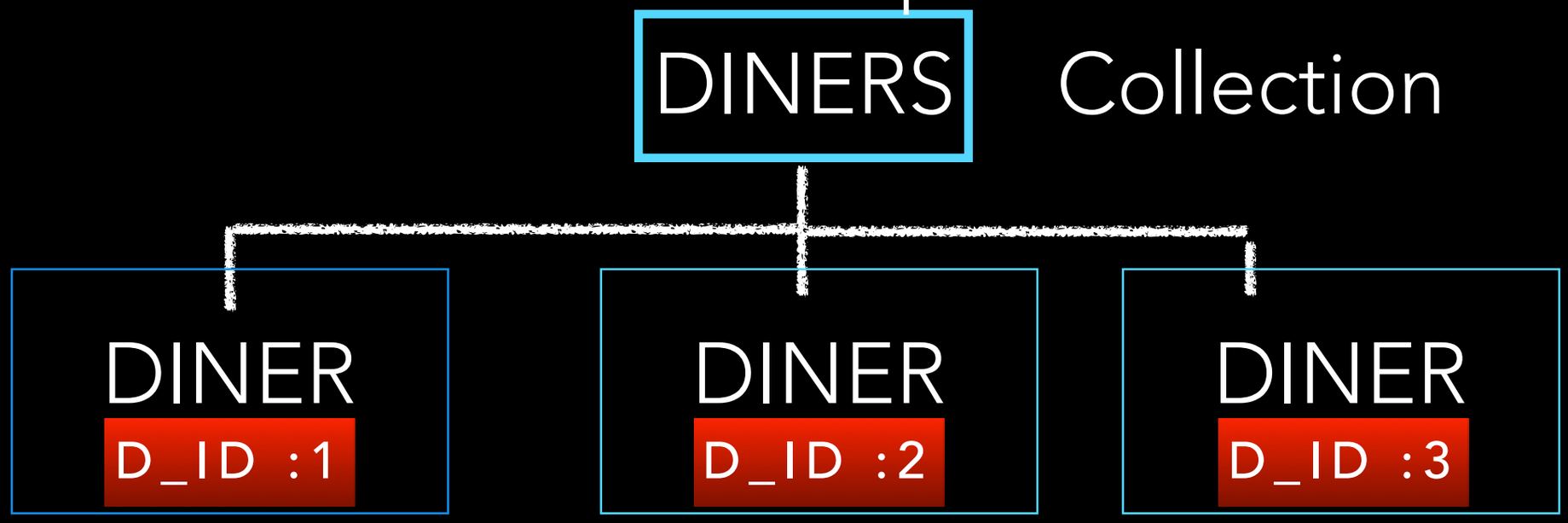
A separate call

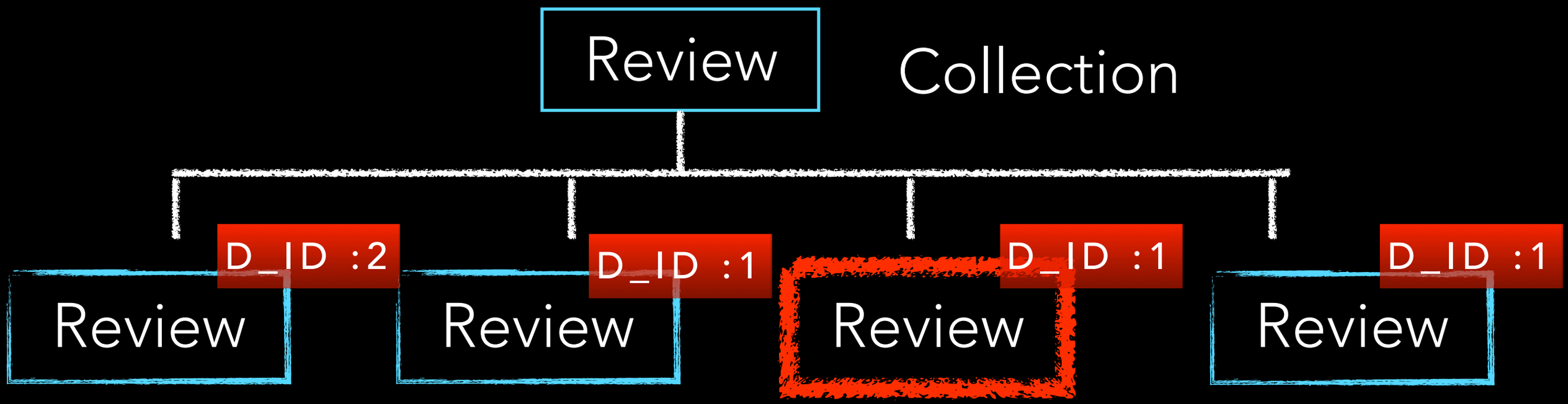




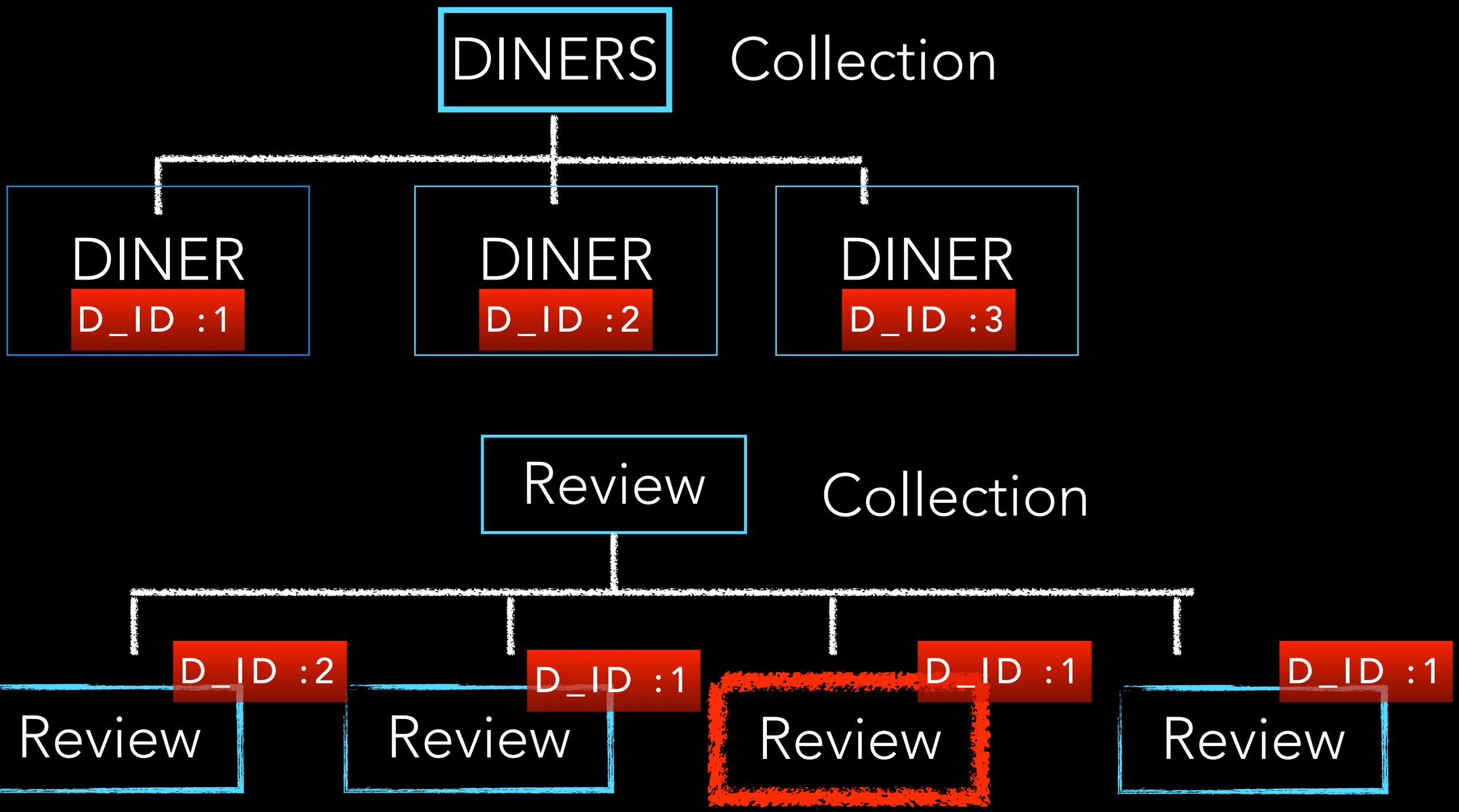
CAN'T request from 2 sub collections need to make

A separate call

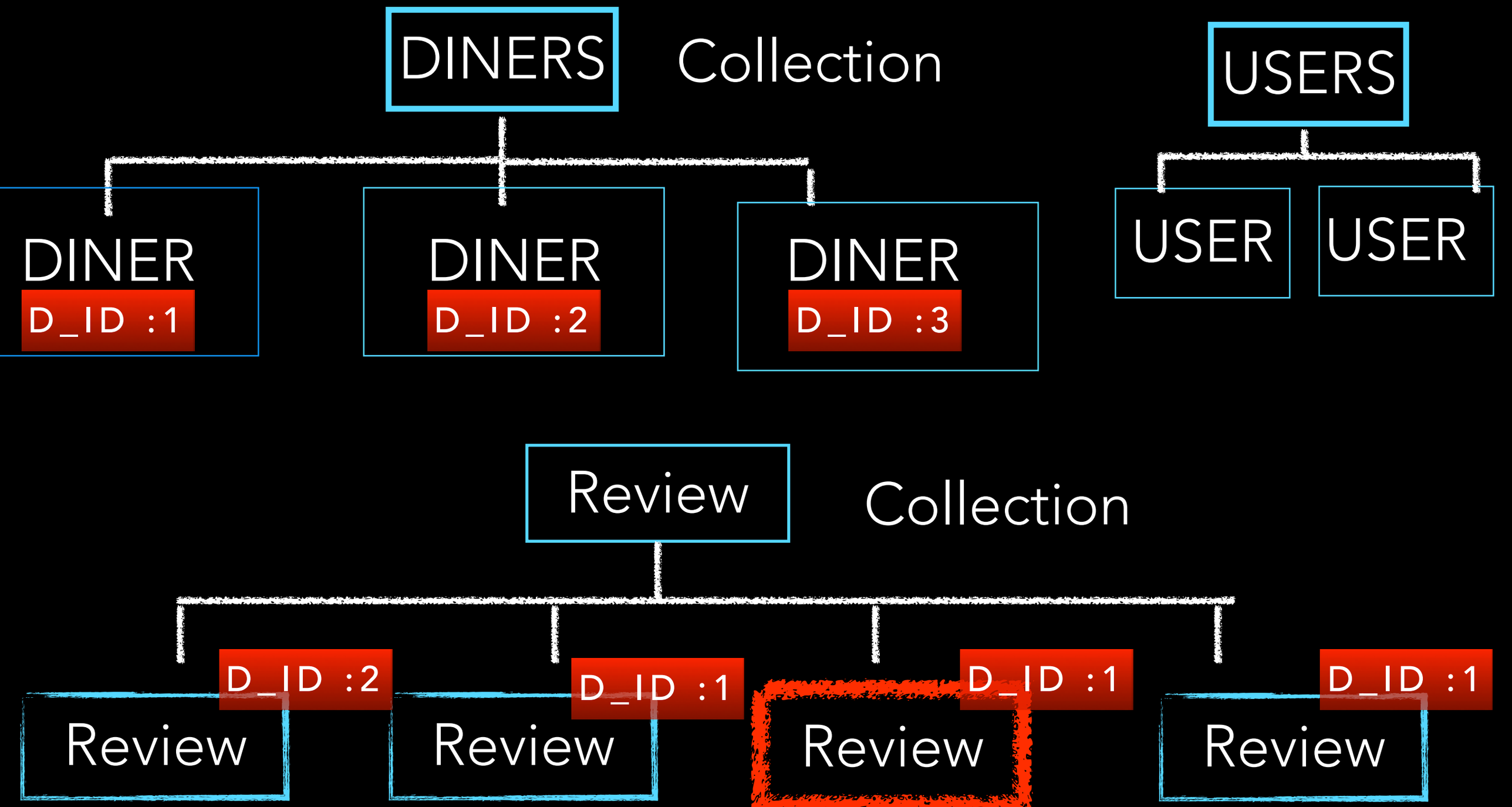




WITH THIS ARCHITECTURE WHAT IF WANT TO SHOW A USERS NAME AND PROFILE



WITH THIS ARCHITECTURE WHAT IF WANT TO SHOW A USERS NAME AND PROFILE



DINERS

DINER
D_ID :1

DINER
D_ID :2

DINER
D_ID :3

USERS

USER
U_ID :1

USER
U_ID :2

WE COULD ADD U_ID BUT THE DATABASE CAN'T AUTOMATICALLY JOIN THEM SO WE HAVE TO A SEPARATE FETCH

SLOW

Review

Review
D_ID :2
U_ID :2

Review
D_ID :1
U_ID :1

Review
D_ID :1
U_ID :1

Review
D_ID :1
U_ID :2

DINERS

DINER
D_ID :1

DINER
D_ID :2

DINER
D_ID :3

USERS

USER

USER

Review

D_ID :2
Review
USER

D_ID :1
Review
USER

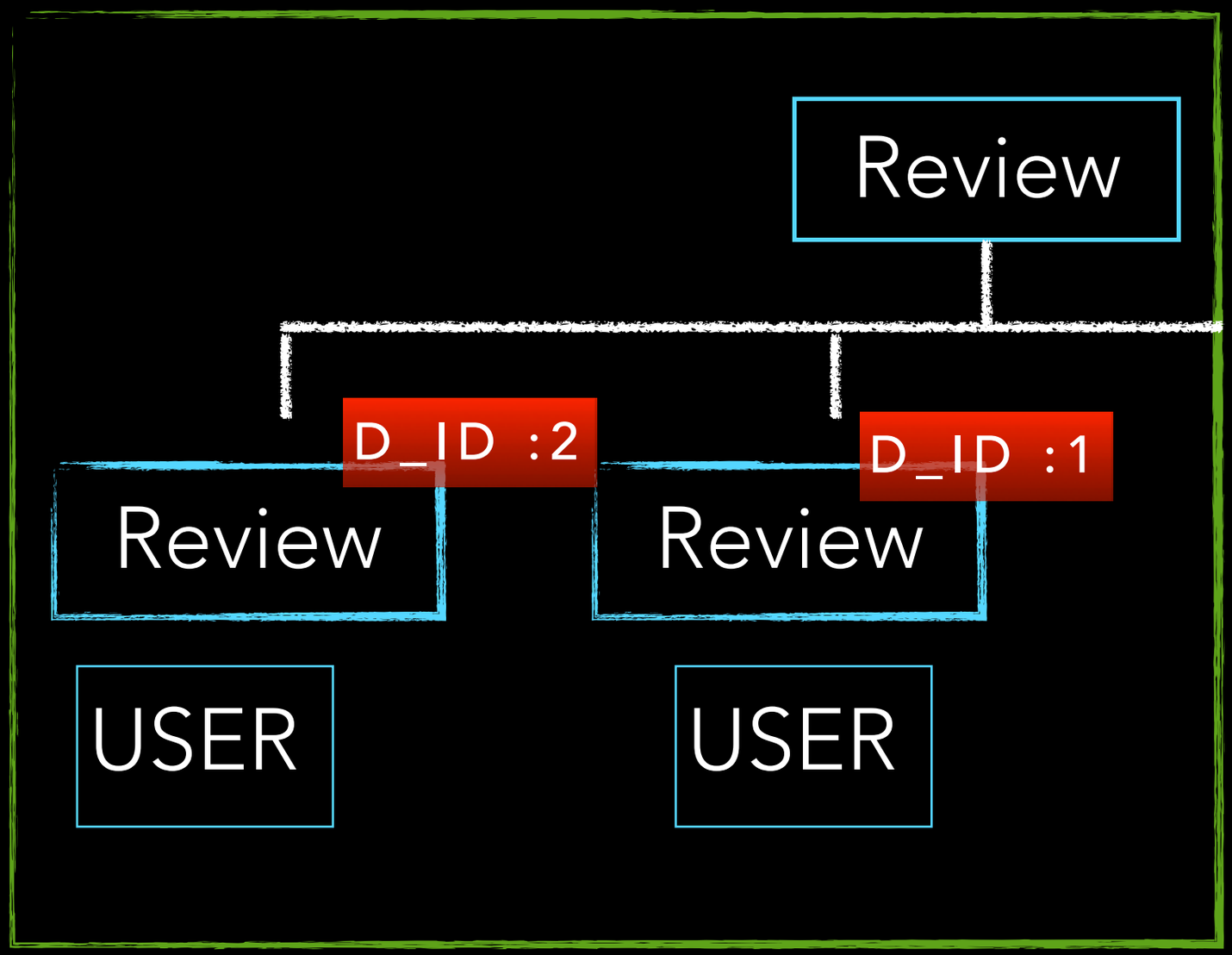
D_ID :1
Review
USER

D_ID :1
Review
USER

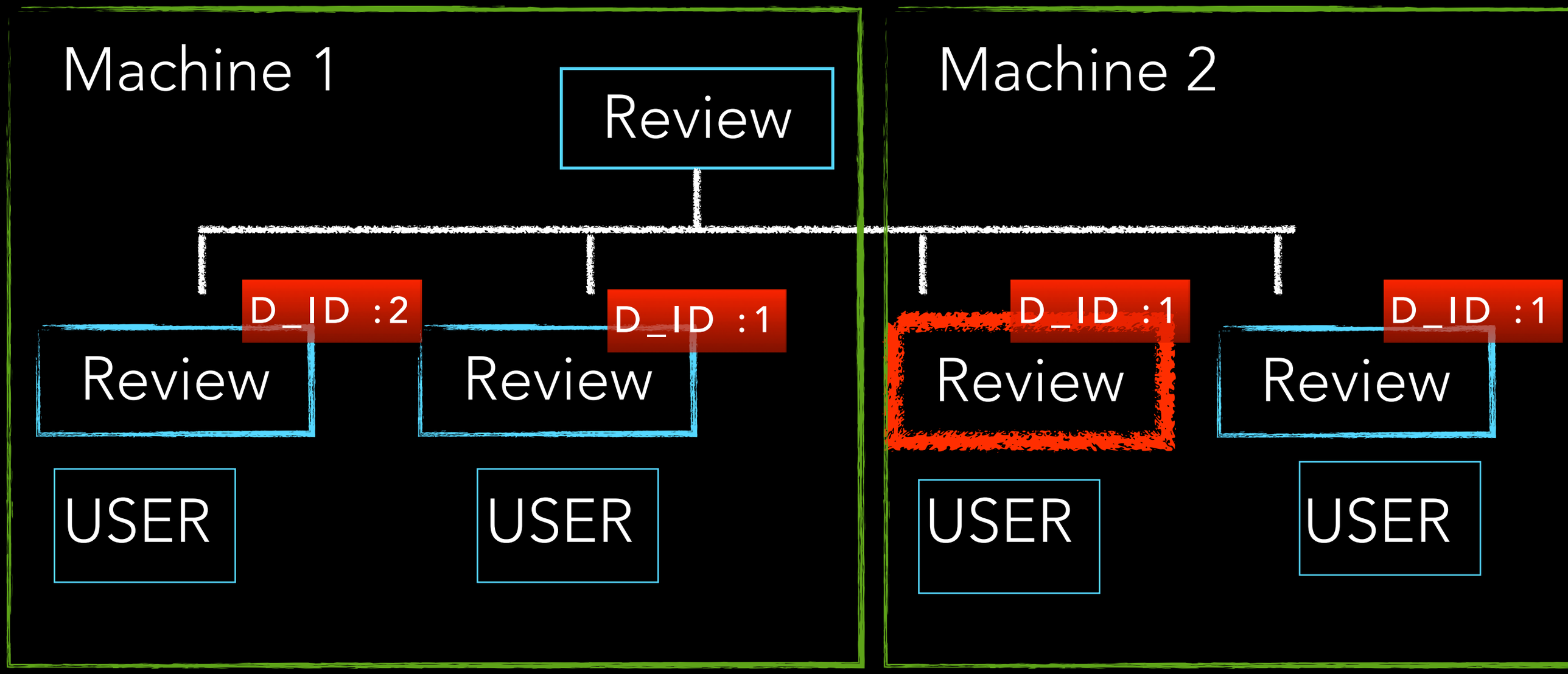
DUPLICATE DATA IS BAD

- You will have update user objects for all review if user data changes.
- Trade offs. (And potential research problem)
    - Since you doing more reads of reviews that updates of user info. It might be worth it.
- Normalizing. No SQL database open question.

DISTRIBUTE INFORMATION ACROSS MULTIPLE MACHINES



DISTRIBUTE INFORMATION ACROSS MULTIPLE MACHINES AND SCALE REAL EASY



STRUCTURE OF FIRESTORE DOCUMENTS

Document

bird_type:
airspeed:
coconut_capacity:
isNative:
icon:
vector:

distances_traveled:

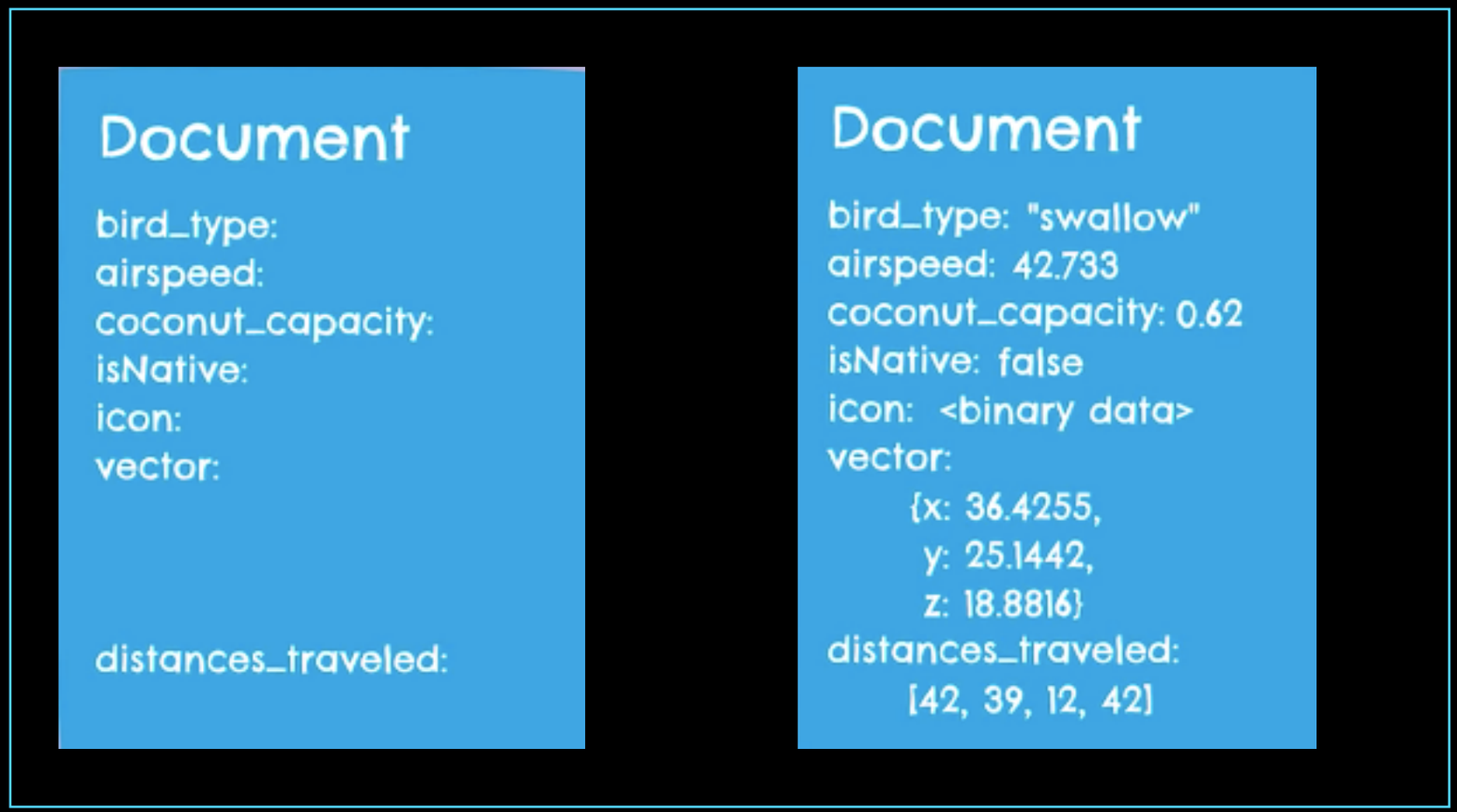
Document

bird_type: "swallow"
airspeed: 42.733
coconut_capacity: 0.62
isNative: false
icon: <binary data>
vector:
{x: 36.4255,
y: 25.1442,
z: 18.8816}
distances_traveled:
[42, 39, 12, 42]

KEY VALUE PAIRS REFERRED TO AS FIELDS

MAPS

STRUCTURE OF FIRESTORE DOCUMENTS



Collection

Rules:

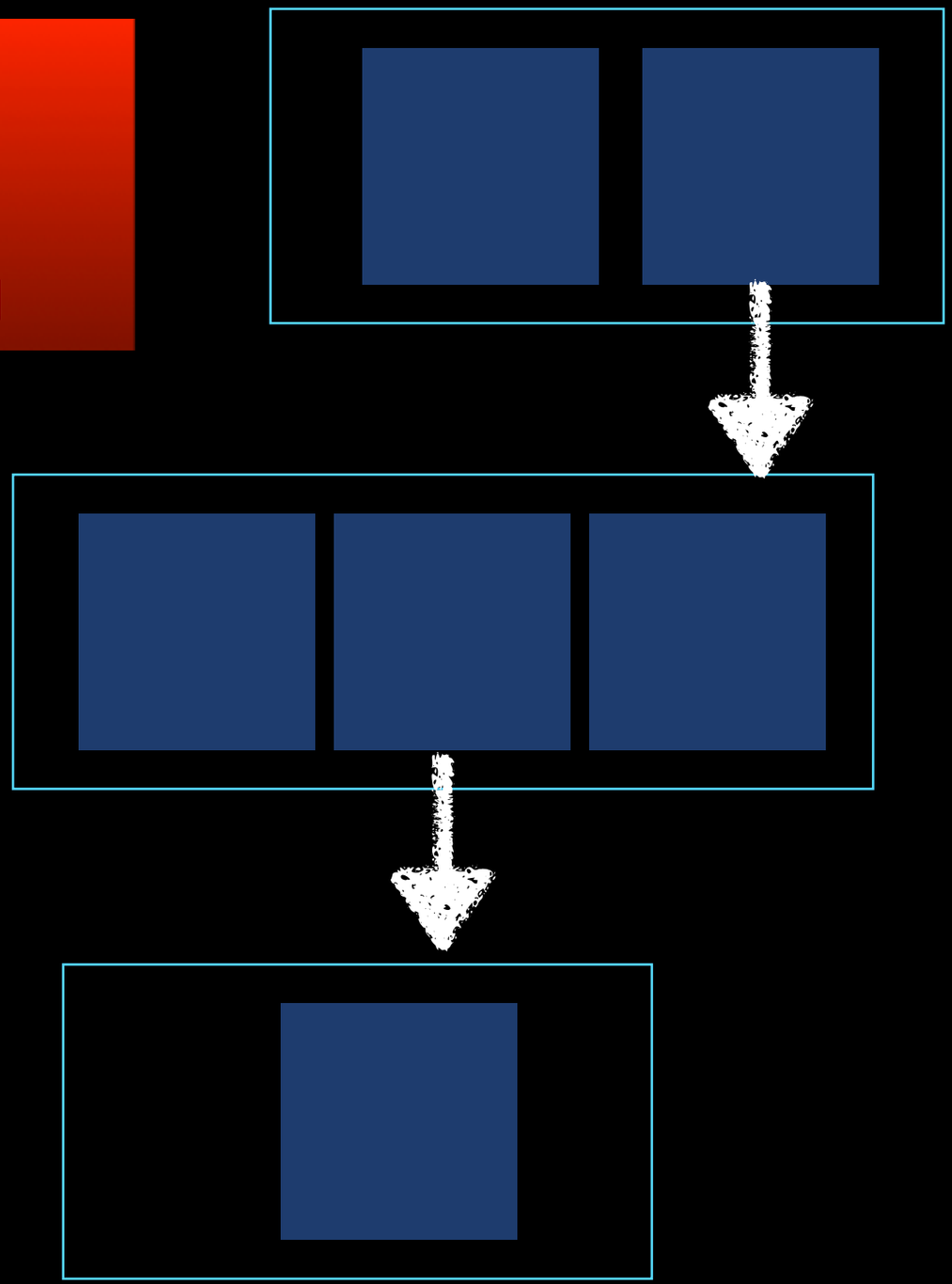
Collection can only contain documents

Document can only be 1MB

Document can point to sub collection

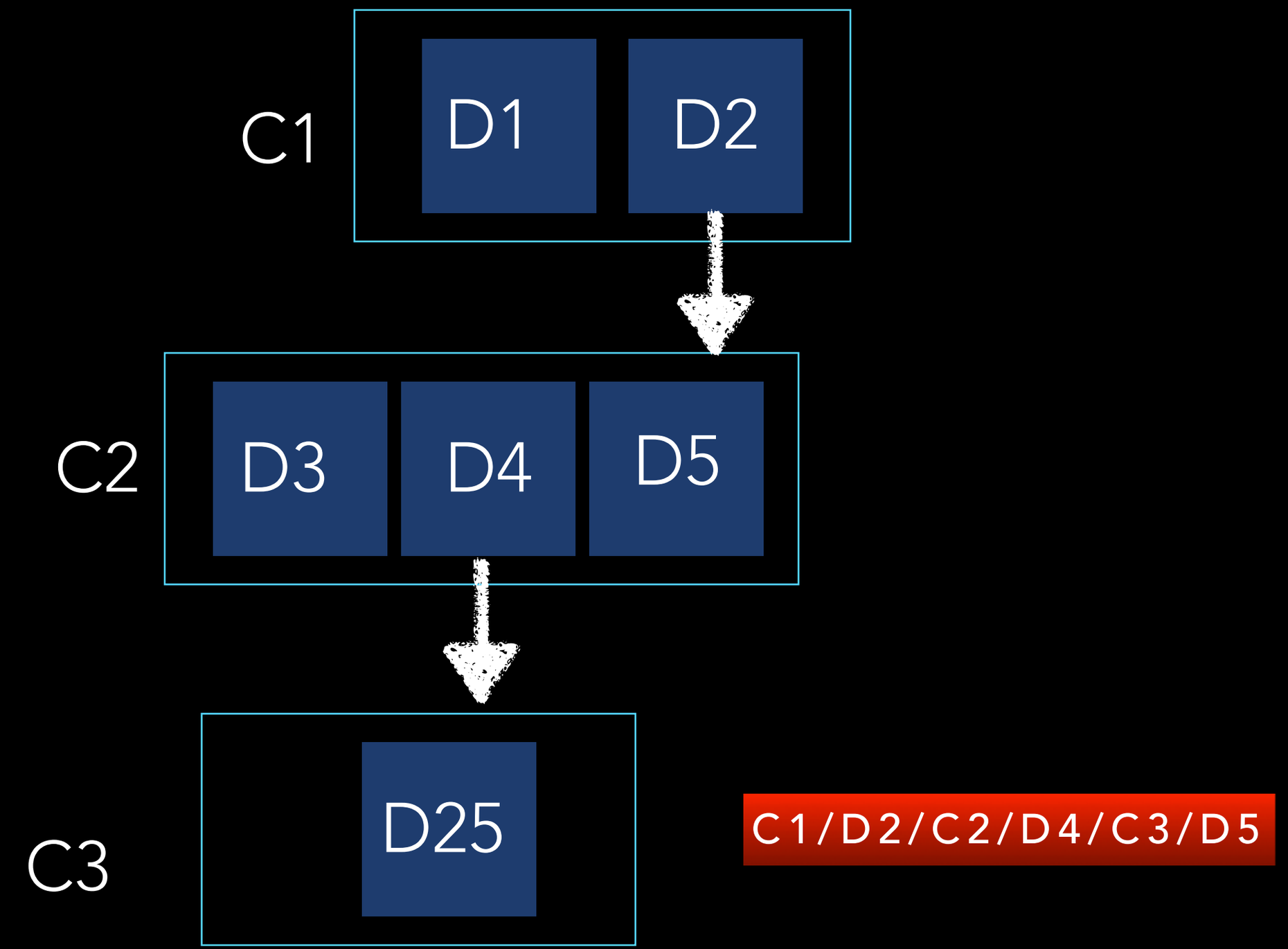
TREE STRUCTURE

THE ROOT CAN ONLY CONTAIN A COLLECTION



TREE STRUCTURE

Path : Collection/Document/Collection/Document



C1/D2/C2/D4/C3/D5

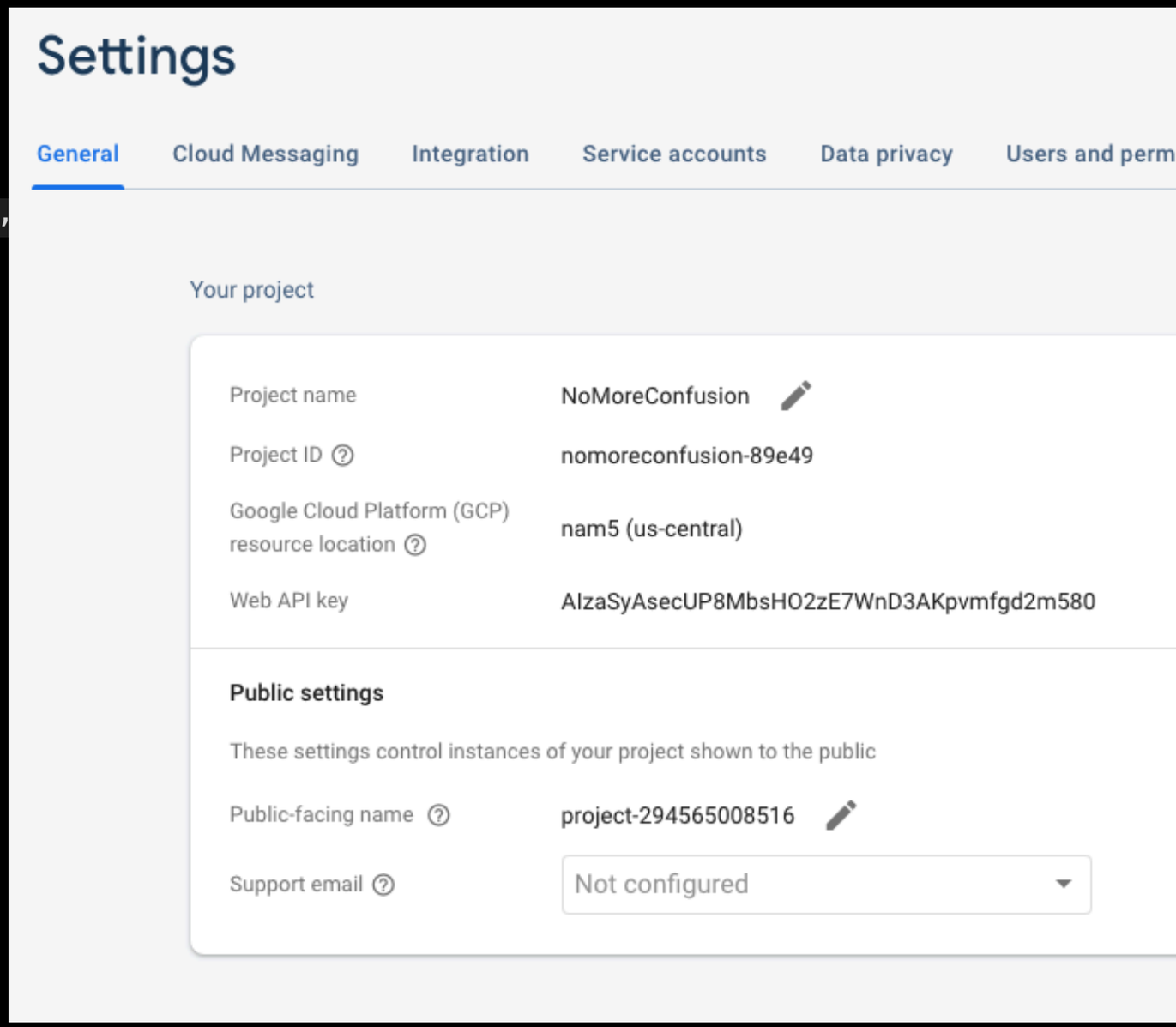
PUSH UPDATES

EXAMPLE APP DEEP DIVE

[HTTPS://NOMORECONFUSION-89E49.FIREBASEAPP.COM/?CLASSID=%22MOBILE%22](https://nomoreconfusion-89e49.firebaseapp.com/?classid=%22mobile%22)

```
var config = {
    apiKey: "AIzaSyAsecUP8MbsHO2zE7WnD3AKpvmfgd2m580",
    authDomain: "nomoreconfusion-89e49.firebaseapp.com",
    databaseURL: "https://nomoreconfusion-89e49.firebaseio.com",
    projectId: "nomoreconfusion-89e49",
    storageBucket: "nomoreconfusion-89e49.appspot.com",
    messagingSenderId: "294565008516"
};
```

https://console.firebase.google.com/u/0/project/_/settings/general/



```
var database = firebase.database()
APP.classID = getParameterByName("classID")
console.log(APP.classID)
var ref = database.ref('feedBack/' + APP.classID)
```

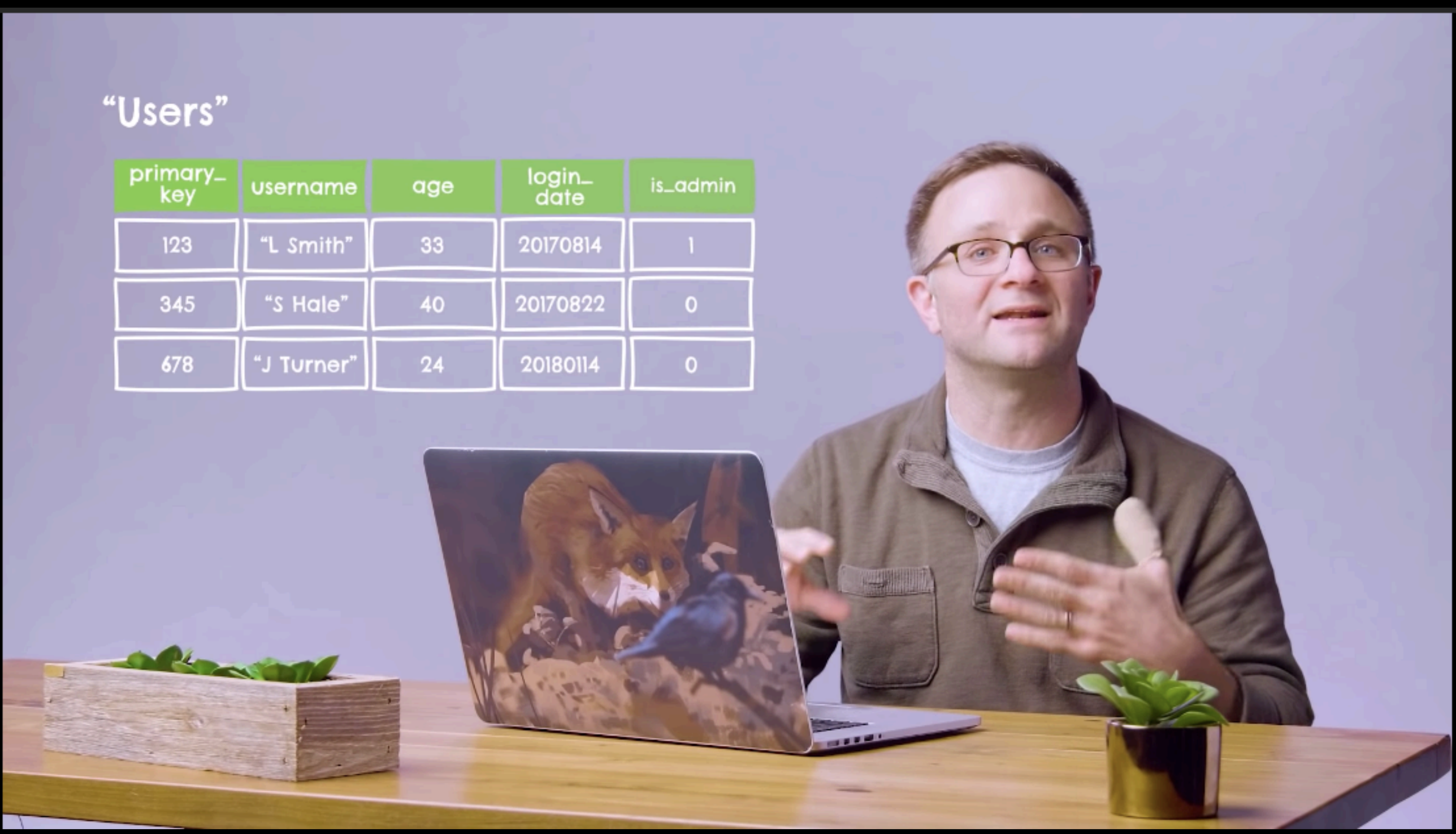
```
function updateCount(numbConfused, numbClear) {

  APP.clear += numbClear
  if(APP.clear < 0){
    APP.clear = 0
  }
  APP.confused += numbConfused
  if(APP.confused <0 ){ APP.confused =0}
  ref.set({
    clear: APP.clear,
    confused: APP.confused,
  })
  console.log("clear " + APP.clear)
  console.log("Confused: " + APP.confused)

}
```

```
ref.on("value", function(snapshot) {
    var updatedEntry = snapshot.val()
      if(updatedEntry !== null){
        setConfused(updatedEntry.confused)
        setClear(updatedEntry.clear)
        console.log("clear " + updatedEntry.clear)
        console.log("Confused: " + updatedEntry.confused)
      }else{
        ref.set({
          clear: 1,
          confused: 0,
        })
      }
    if(APP.first == true ){
      if (checkCookie(APP.cookieName) == false){
        spu_createCookie(APP.cookieName, "today", 2)
        updateCount(0,1)
      }

      APP.first = false
      document.body.style.backgroundColor = "#efe0b1"

    }

  })
```

WHAT IS FIREBASE

"Users"

primary_key	username	age	login_date	is_admin
123	"L Smith"	33	20170814	1
345	"S Hale"	40	20170822	0
678	"J Turner"	24	20180114	0



HTTPS://WWW.YOUTUBE.COM/WATCH?V=V_HR4K4AUOQ