

Firestore queries

Section 11

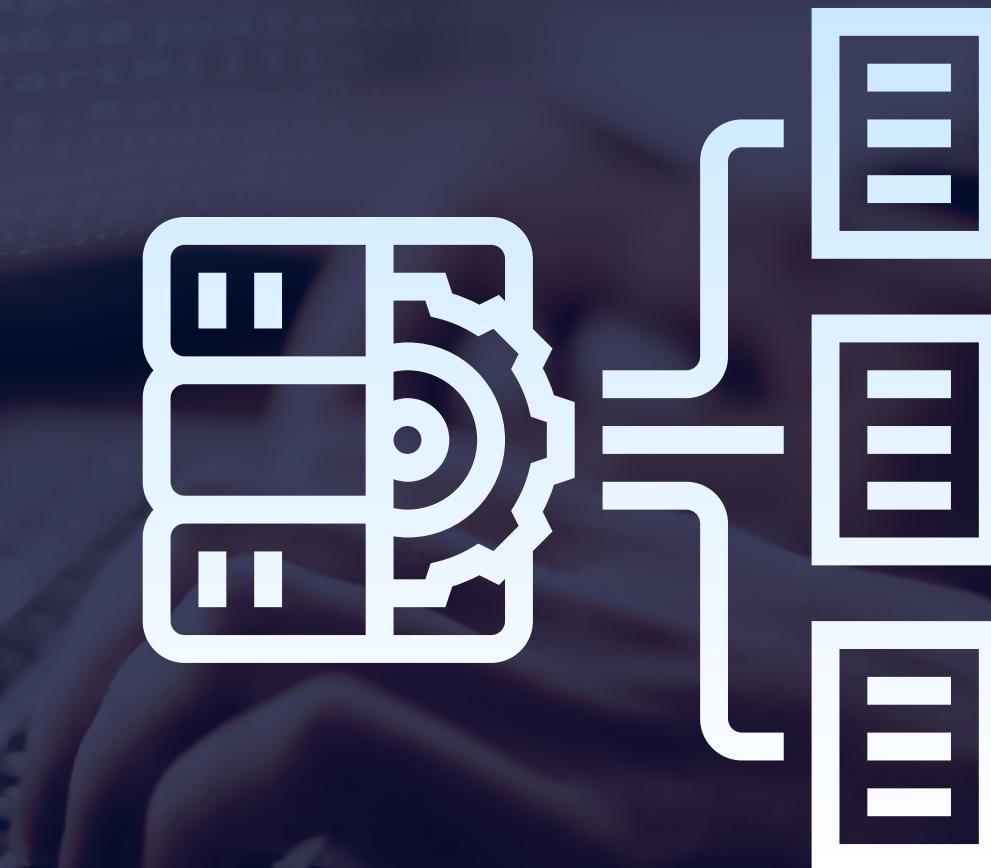
In this section

- Firestore DB design
 - Firestore “Relationships”
 - User event signup



The goal of this section

Firestore database design

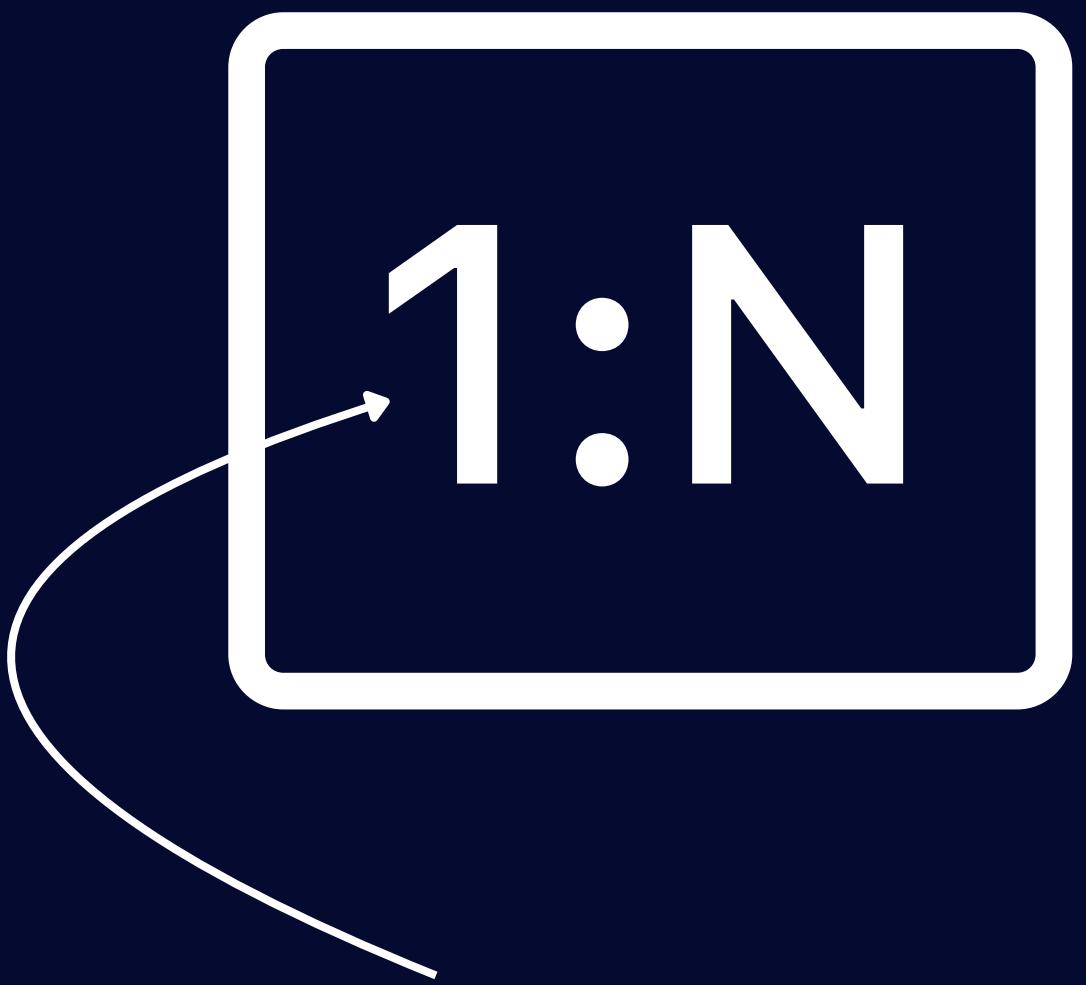


events > activity-f1

More in Google Cloud

(default)	events	activity-f1
+ Start collection	+ Add document	+ Start collection
events >	activity-f1 >	+ Add field
profiles	activity-f10	attendeeIds
	activity-f11	0 "jane-id"
	activity-f12	1 "bob-id"
	activity-f13	attendees
	activity-f14	0
⋮: activity-f2	activity-f3	displayName: "Jane"
	activity-f4	id: "jane-id"
	activity-f5	isHost: true
	activity-f6	photoURL: "https://randomuser.me/api/portraits/women/8.jpg"
	activity-f7	1
	activity-f8	displayName: "Bob"
	activity-f9	id: "bob-id"
	activity-p1	photoURL: "https://randomuser.me/api/portraits/men/1.jpg"
	activity-p2	category: "culture"
		city: "London"
		date: May 1, 2025, 10:00:16 AM UTC+17

Event and user relationship



1 user can attend many events



1 event can have many attendees (users)

Event and user relationship



Many to many relationship between events and users

How it would look in SQL

Events

ID	String
Title	varchar
Date	date

Users

ID	String
Email	varchar
PhotoURL	varchar

EventAttendees

ID	String
UID	string
EventID	string

EventId + UID

How it would look in SQL

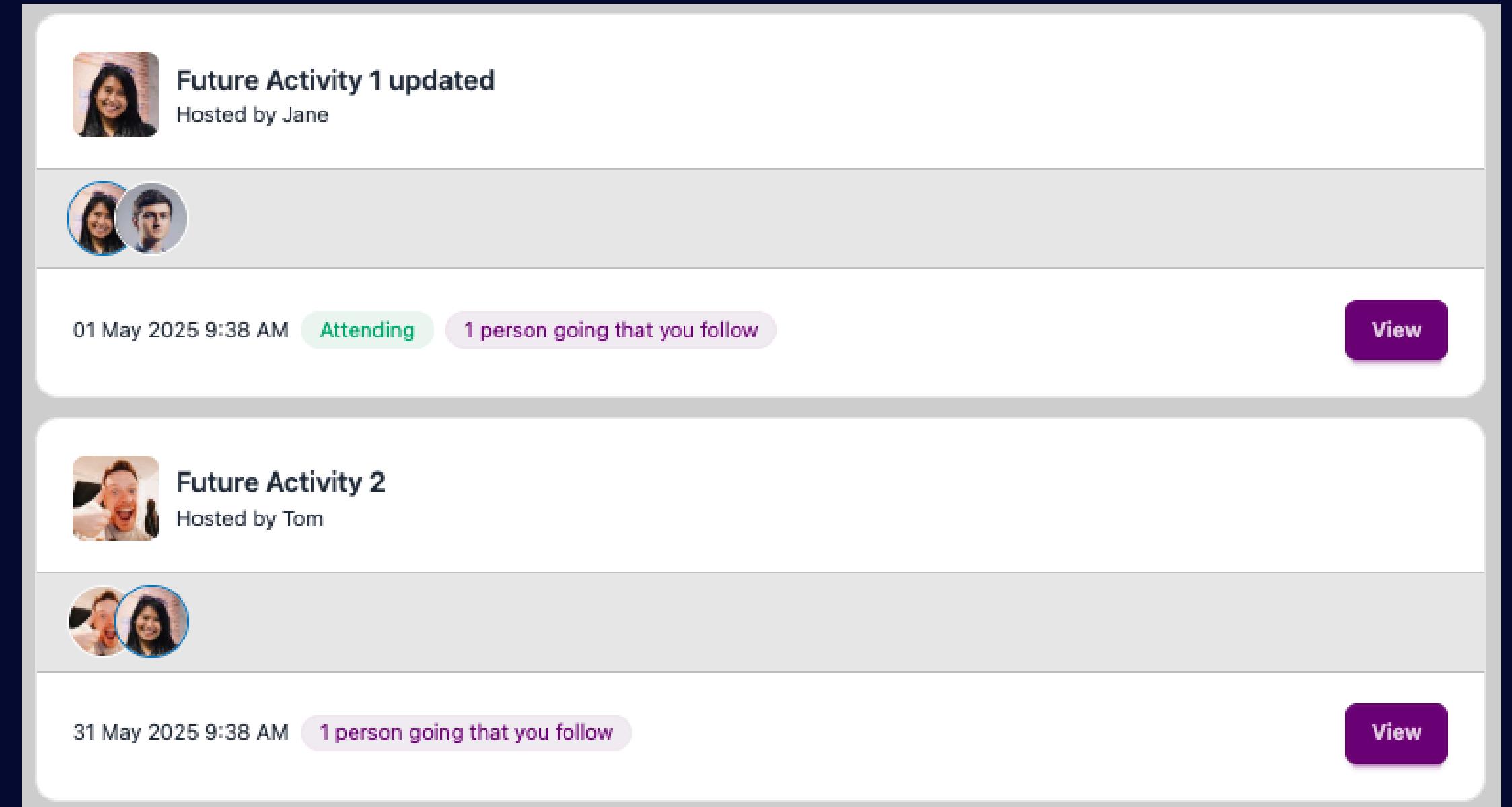
```
SELECT event.Name AS EventName,  
       event.Date AS EventDate,  
       user.Name AS AttendeeName  
  FROM Events AS event  
INNER JOIN Attendees AS a ON event.Id = a.EventId  
INNER JOIN Users AS user ON user.UId = a.UId  
 WHERE event.Id = 4;
```

Firestore - design by the queries

User is going

User is hosting

Date order



events > activity-f1

More in Google Cloud

(default)	events	activity-f1
+ Start collection	+ Add document	+ Start collection
events >	activity-f1 >	+ Add field
profiles	activity-f10	attendeeIds
	activity-f11	0 "jane-id"
	activity-f12	1 "bob-id"
	activity-f13	attendees
	activity-f14	0
⋮: activity-f2	activity-f3	displayName: "Jane"
	activity-f4	id: "jane-id"
	activity-f5	isHost: true
	activity-f6	photoURL: "https://randomuser.me/api/portraits/women/8.jpg"
	activity-f7	1
	activity-f8	displayName: "Bob"
	activity-f9	id: "bob-id"
	activity-p1	photoURL: "https://randomuser.me/api/portraits/men/1.jpg"
	activity-p2	category: "culture"
		city: "London"
		date: May 1, 2025, 10:00:16 AM UTC+17

Firestore queries



Order by

```
const q = query(  
  eventsRef,  
  orderBy("date")  
);
```

All events in date order

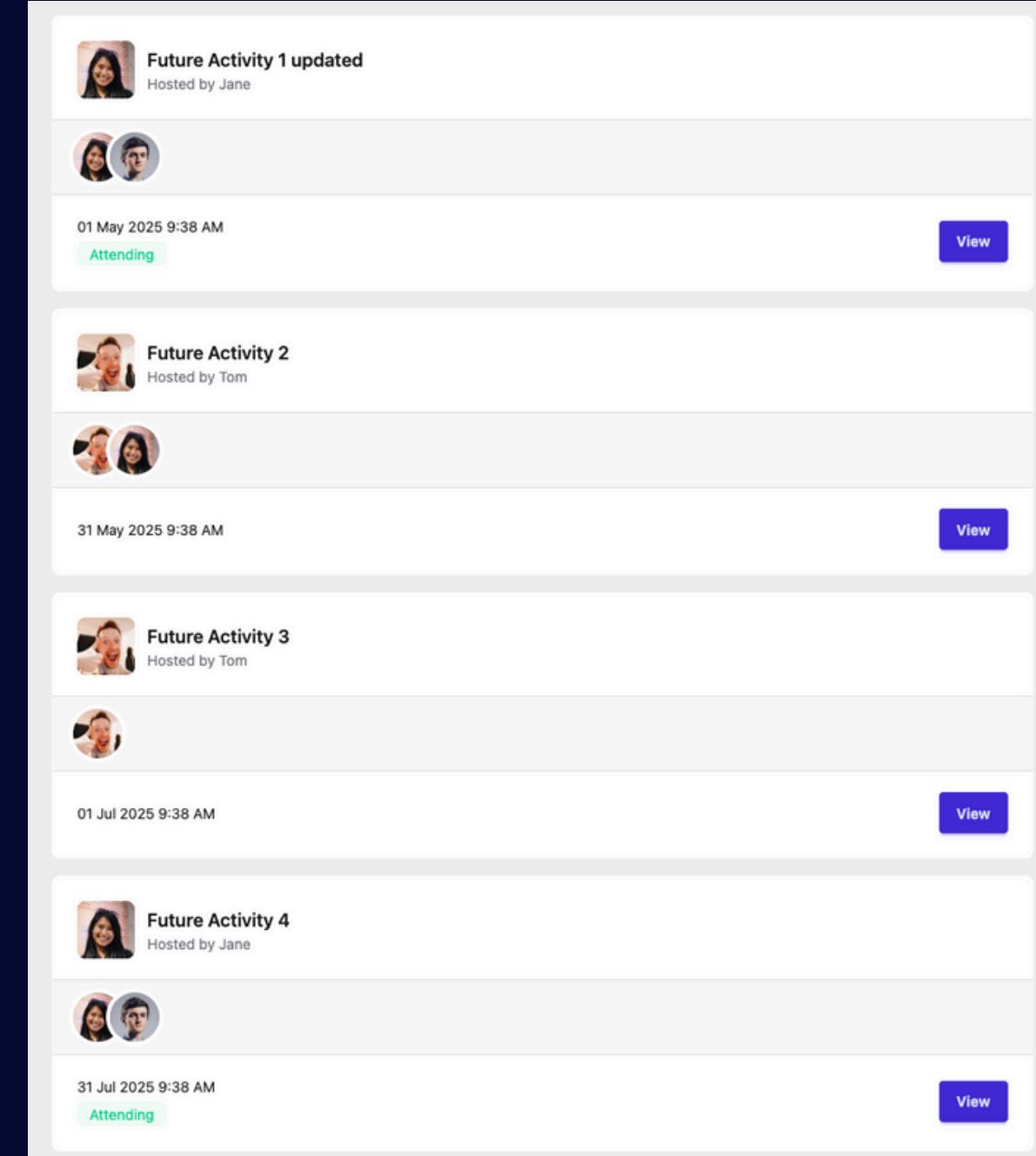
 Past Activity 1 Hosted by Bob	31 Jan 2025 9:38 AM	View
		
	03 Mar 2025 9:38 AM	View
 Past Activity 2 Hosted by Tom	Attending	
		
	01 May 2025 9:38 AM	View
 Future Activity 1 updated Hosted by Jane	Attending	
		
	31 May 2025 9:38 AM	View
 Future Activity 2 Hosted by Tom		
		



Filter + ordering

All events in date order,
filter out past events

```
const q = query(  
  eventsRef,  
  where('date', '>=', Timestamp.now())  
  orderBy("date")  
) ;
```



Compound



```
const q = query(
  eventsRef,
  where('attendeeIds', 'array-contains', 'bob-id'),
  where('date', '>=', Timestamp.now())
  orderBy("date")
);
```



This type of query requires index

The screenshot shows a mobile application interface. At the top, there's a navigation bar with three colored dots (red, yellow, green) and a search bar. Below the navigation is a code block demonstrating a Firestore query. A white curved arrow points from this code block upwards towards the database interface. The main content area displays a list of events under the heading "Events". Each event card includes a thumbnail image, the event name (e.g., "Future Activity 1 updated"), the location ("Natural History Museum"), and the date ("01 May 2025 9:38 AM"). Below the event list is a detailed view of attendees. The attendee list starts with "attendeeIds" containing "jane-id" and "bob-id". Under "attendees", there are two entries: "0" (Jane) and "1" (Bob). Each attendee has a "displayName", "id", "isHost" (set to true for Jane), and a "photoURL". At the bottom right of the attendee list is a "Add field" button.

+ Add field

- attendeeIds

0 "jane-id"

1 "bob-id"

- attendees

- 0

displayName: "Jane"

id: "jane-id"

isHost: true

photoURL: "https://randomuser.me/api/portraits/women/8.jpg"

- 1

displayName: "Bob"

id: "bob-id"

photoURL: "https://randomuser.me/api/portraits/men/1.jpg"

Add field

(map) + Delete

Events

Future Events Past Events Hosting

Future Activity 1 updated

Natural History Museum
01 May 2025 9:38 AM

Future Activity 4

The Blackfriar
31 Jul 2025 9:38 AM

Future Activity 5

Metropolitan Museum of Art
31 Aug 2025 9:38 AM

Future Activity 6

Tokyo Dome
01 Oct 2025 9:38 AM

Future Activity 8

Colosseum
01 Dec 2025 9:38 AM

Future Activity 9

Berghain
31 Dec 2025 9:38 AM

Firestore - duplicating the data



Bob
bob-id

Event1

Event2

Event3

Firestore - duplicating the data

To get Events + users attending

Get Events = 3 doc reads

Event1



Bob
bob-id

Event2



Bob
bob-id

Event3



Bob
bob-id

Firestore - if we didn't duplicate



Bob

bob-id

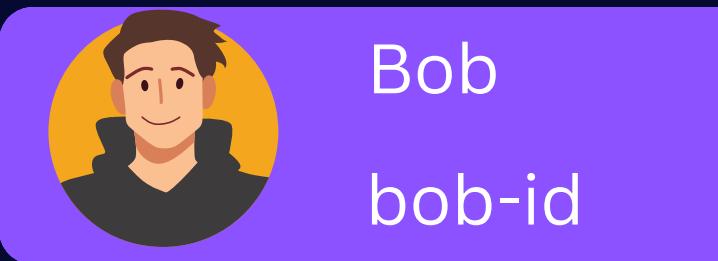
Event1

Event2

Event3

Firestore - if we didn't duplicate

1 Query for each event (3 doc reads)



Event1

bob-id

Event2

bob-id

Event3

bob-id

1 Query for each attendee in each event

Data consistency?

Summary

In this section

- Firestore DB design
- Firestore “Relationships”





FAQ



So, duplicate data. If a user signs up to 100 events does that mean I have to change the attendees details in every event?



Yes. This is typical for a NoSQL DB

Business decision about what “should” be updated as it requires batch writes and potential cost.

Up next

