Collection -> doc (document)->data.

Collection chứa nhiều doc = db.collection('rooms').doc('roomA')  
                .collection('messages').doc('message1');

Read firebase =get()

db.collection("users").get().then((querySnapshot) => {  
    querySnapshot.forEach((doc) => {  
        console.log(`${doc.id} => ${doc.data()}`);  
    });  
});

Add firebase = set()

* db.collection(‘user’).doc(‘infor).set({

first\_name:’’,

last\_name:’’

}).then(res=>{

}).catch(err=>{

})

* Var user={

First\_name:’’,

Last\_name:’’,

Age:16

};

Db.collection(“data”).doc(“class”).set(“user”).then(…)

Add a document :

// Add a new document with a generated id.  
db.collection("cities").add({  
    name: "Tokyo",  
    country: "Japan"  
})  
.then(function(docRef) {  
    console.log("Document written with ID: ", docRef.id);  
})  
.catch(function(error) {  
    console.error("Error adding document: ", error);  
});

.add(…) = .doc().set(… )

Update a document:

var washingtonRef = db.collection("cities").doc("DC");  
  
// Set the "capital" field of the city 'DC'  
return washingtonRef.update({  
    capital: true  
})  
.then(function() {  
    console.log("Document successfully updated!");  
})  
.catch(function(error) {  
    // The document probably doesn't exist.  
    console.error("Error updating document: ", error);  
});

Update fields in nested objects:

// Create an initial document to update.  
var frankDocRef = db.collection("users").doc("frank");  
frankDocRef.set({  
    name: "Frank",  
    favorites: { food: "Pizza", color: "Blue", subject: "recess" },  
    age: 12  
});  
  
// To update age and favorite color:  
db.collection("users").doc("frank").update({  
    "age": 13,  
    "favorites.color": "Red"  
})  
.then(function() {  
    console.log("Document successfully updated!");  
});

Bath (ghi hàng loạt)

// Get a new write batch  
var batch = db.batch();  
  
// Set the value of 'NYC'  
var nycRef = db.collection("cities").doc("NYC");  
batch.set(nycRef, {name: "New York City"});  
  
// Update the population of 'SF'  
var sfRef = db.collection("cities").doc("SF");  
batch.update(sfRef, {"population": 1000000});  
  
// Delete the city 'LA'  
var laRef = db.collection("cities").doc("LA");  
batch.delete(laRef);

Delete documents:

Db.collection(“users”).db(“info”).delete().then(…).catch(…)

**Warning:** Deleting a document does not delete its subcollections!

Delete fields:

var cityRef = db.collection('cities').doc('BJ');  
  
// Remove the 'capital' field from the document  
var removeCapital = cityRef.update({  
    capital: firebase.firestore.FieldValue.delete()  
});

db.collectiion(“school”).doc(“class”)

+Update: sẽ update collection và bị lổi nếu doc() không tồn tại

+Set (Not mergeFields) :sẽ ghi dè vào một doc() hoặc tạo mới nếu nó không tồn tại

+Set(mergeField):sẽ cập nhật doc() hoặc tạo mới nó nếu nó khong tồn tại

+Create : sẽ tạo một doc() bị lỗi nếu doc() đã tồn tại