**IV. Implementation of Relation Model via MySQL and NoSQL**

**TABLE CREATION AND INSERTION (MYSQL)**

**Question:** Write a query to create a holder health history table having holder id as primary key, and other attributes like health history of disease, the id of hospital and the last date of disease. Also, perform data insertion into the table.

**Answer:**

use insurance;

CREATE TABLE policy\_holder\_health\_history (

Holder\_id INT NOT NULL PRIMARY KEY,

Health\_history\_disease VARCHAR(255),

Health\_history\_hospital\_id INT,

Health\_history\_date VARCHAR(255)

);

select \* from policy\_holder\_health\_history;

INSERT INTO policy\_holder\_health\_history

(Holder\_id,Health\_history\_disease,Health\_history\_hospital\_id,Health\_history\_date)

VALUES

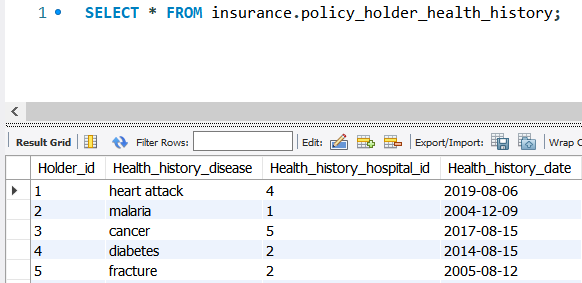
(1,'heart attack',4,'2019-08-06'),

(2,'malaria',1,'2004-12-09'),

(3,'cancer',5,'2017-08-15'),

(4,'diabetes',2,'2014-08-15'),

(5,'fracture',2,'2005-08-12');



**DATA RETRIEVAL USING SUB-QUERIES (MYSQL)**

**Question:** Display the policy holder id, name and amount of installment paid every cycle of an individual who pays the highest installment amount yearly from the insurance database.

**Answer:**

select c.Holder\_id, p.name,c.Amount\_of\_installmentcycle

from policyholder p, policyholdercycle c

where p.holderid=c.holder\_id and c.Amount\_of\_installmentcycle in

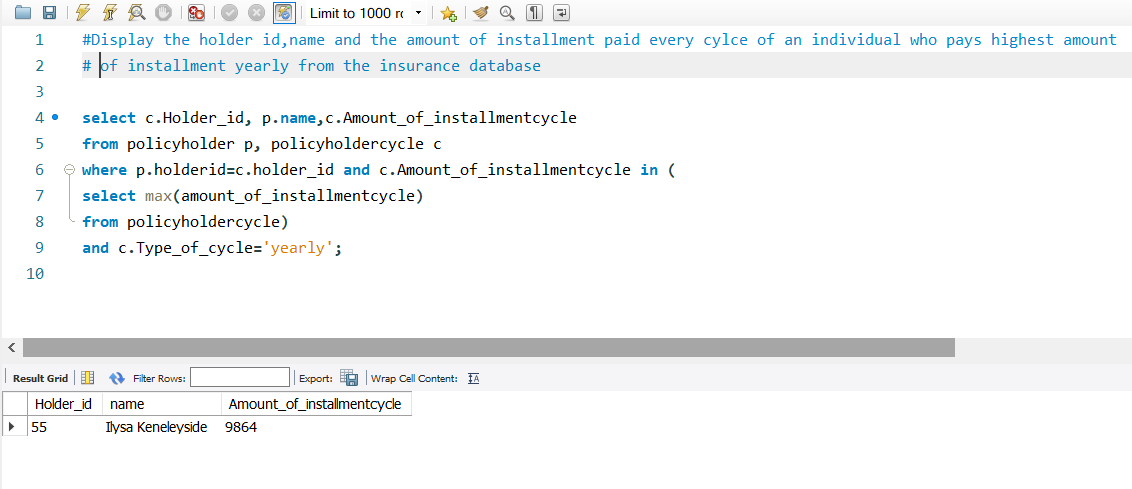
(

select max(amount\_of\_installmentcycle)

from policyholdercycle

)

and c.Type\_of\_cycle='yearly';



**QUERY USING AGGREGATE FUNCTIONS (MYSQL)**

**Question:** Display maximum claim amount given to an individual and the average claim amount given by the insurance company per individual.

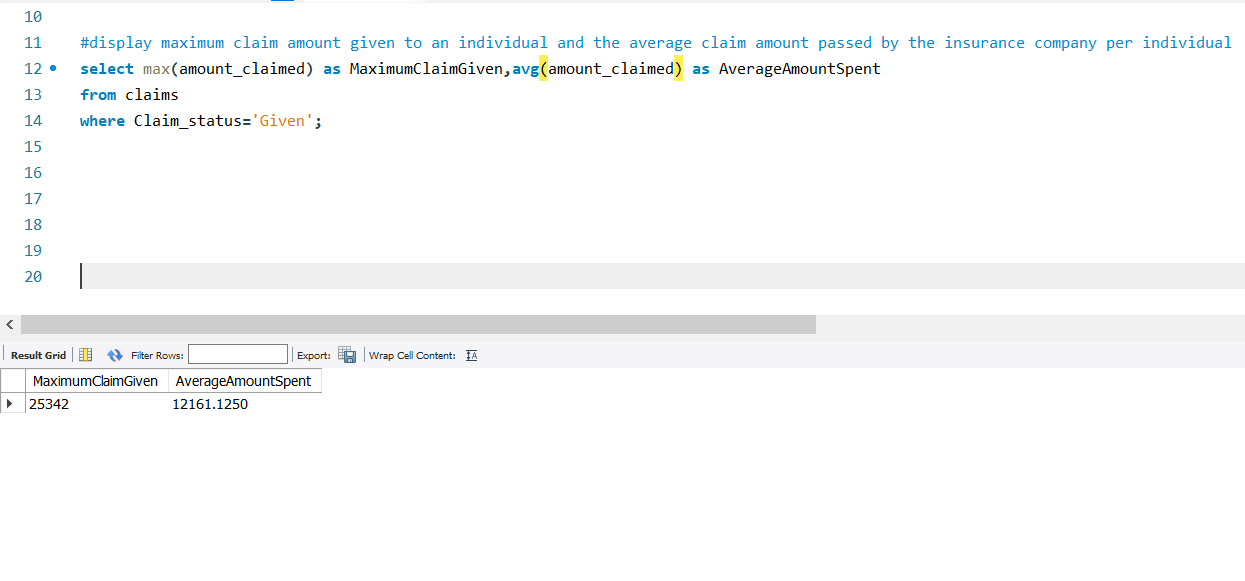
**Answer:**

select max(amount\_claimed) as MaximumClaimGiven,

avg(amount\_claimed) as AverageAmountSpent

from claims

where Claim\_status='Given';

****

**DATA RETRIEVAL USING JOINS ON TWO TABLES (MYSQL)**

**Question :** Display the Policy holder id, name, age, insurance id and health history of the individuals born after 1900.

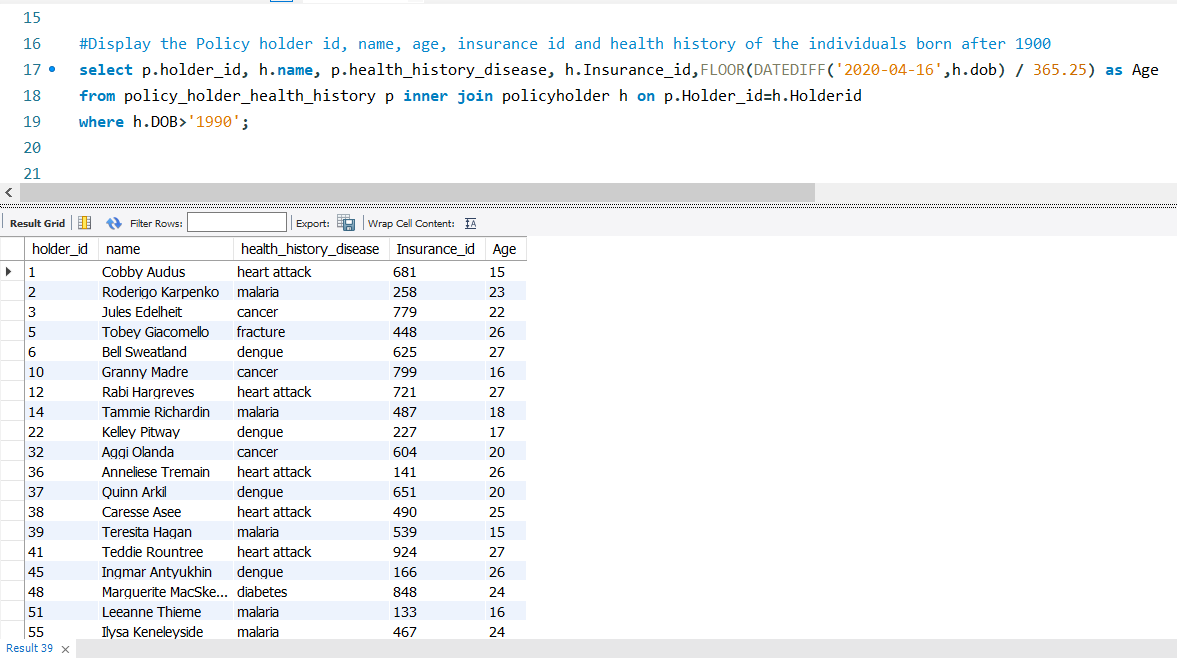
**Answer:**

select p.holder\_id, h.name, p.health\_history\_disease,

h.Insurance\_id, FLOOR(DATEDIFF('2020-04-16',h.dob) / 365.25) as Age

from policy\_holder\_health\_history p inner join policyholder h on p.Holder\_id=h.Holderid

where h.DOB>'1990';



**VIEW CREATION USING AGGREGATE FUNCTIONS, PERFORMING MULTIPLE JOINS ON TABLES AND DATA RETRIEVAL USING GROUP BY (MYSQL)**

**Question:** Create a view and display details of policy holders(id,name,previous health history,age) who paid maximum money to the company for every type of cycle and their difference from average amount. Also, print the minimum of each cycle.

**Answer:**

create view v as(

select p.holder\_id, h.name, p.health\_history\_disease, h.Insurance\_id,

FLOOR(DATEDIFF('2020-04-16',h.dob) / 365.25) as Age,

c.Amount\_of\_installmentcycle, c.Type\_of\_cycle

from policy\_holder\_health\_history p inner join policyholder h on p.Holder\_id=h.Holderid

inner join policyholdercycle c on c.Holder\_id=h.Holderid

);

select holder\_id,name,age,health\_history\_disease as previousDisease,

max(amount\_of\_installmentcycle) as Amount\_Paid\_by\_holder, type\_of\_cycle,

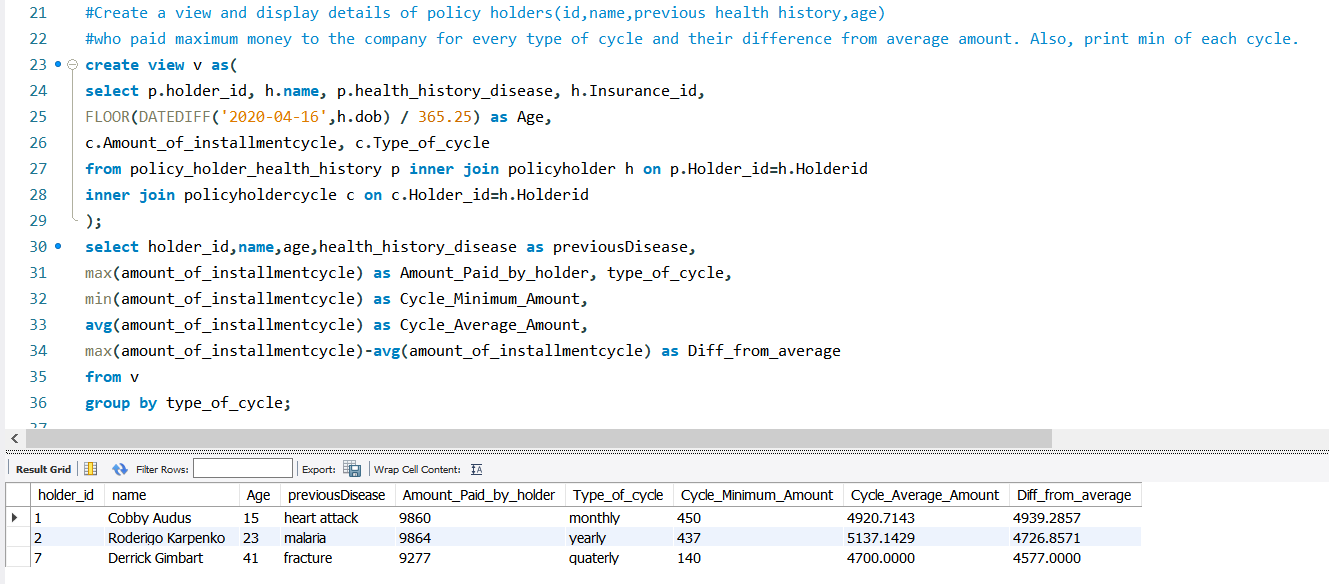
min(amount\_of\_installmentcycle) as Cycle\_Minimum\_Amount,

avg(amount\_of\_installmentcycle) as Cycle\_Average\_Amount,

max(amount\_of\_installmentcycle)-avg(amount\_of\_installmentcycle) as Diff\_from\_average

from v

group by type\_of\_cycle;

****

For **NoSQL**, we used 2 tables in MongoDB, one is “claim”, which recorded all the claims that policyholders requested for. Second one is “policy holder”, this records all the details of each holder.

**TABLE CREATION AND INSERTION (NOSQL)**

**Question:** Write a query to create and insert data in a claim table having claim id as primary

key, and other attributes like insurance id of the policy holder, claim status and total amount in the claim.

**Answer:**

db.claim.insert({

Claimid: 1 ,

Insurance\_id: 233,

Claim\_status: "Claimed",

Amount\_claimed: 28594

})

db.claim.insert({

Claimid: 2 ,

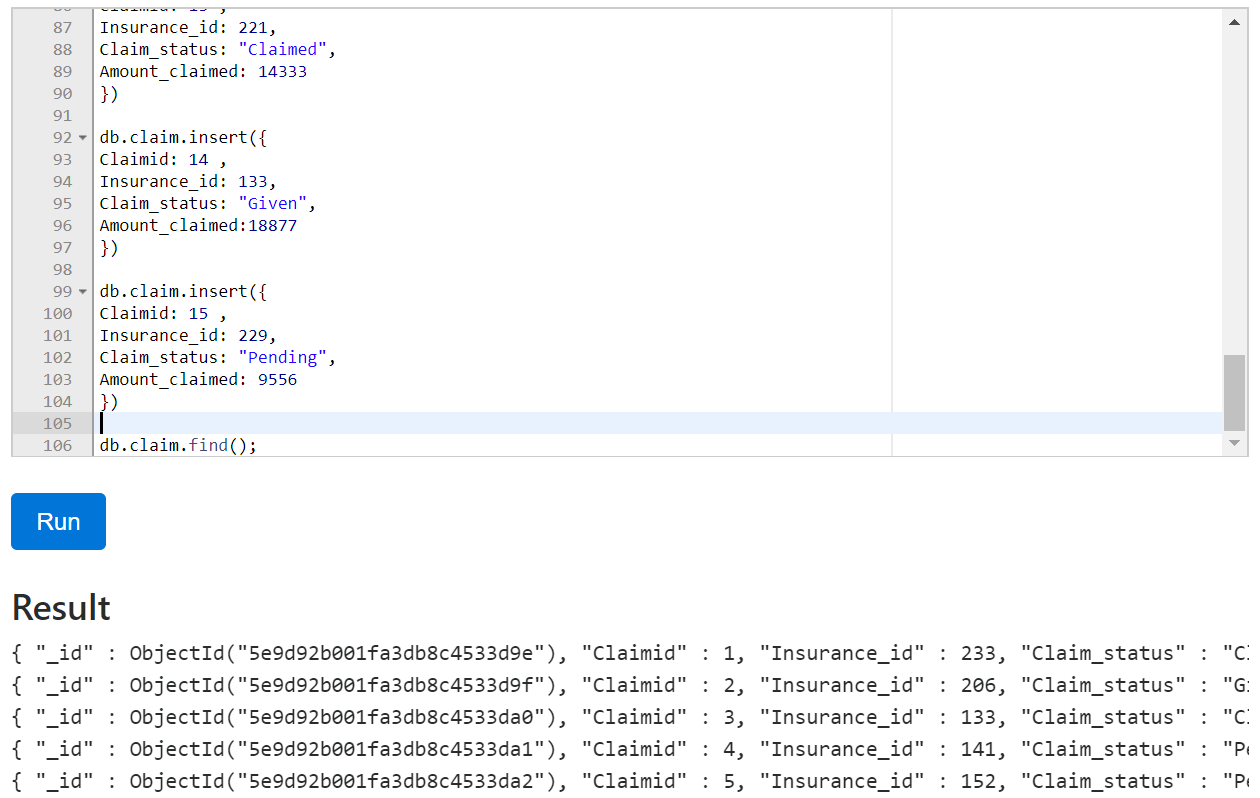
Insurance\_id: 206,

Claim\_status: "Given",

Amount\_claimed: 20220

})

⋮



**BASIC QUERY IN MANGODB**

**Question:** List out all the policyholders that never received the bill.

**Answer:**

db.policyholder.find({ "Bill\_id": null });



**AGGREGATE PIPELINE**

**Question:** Write a query to get the highest amount in the given claim using an aggregate pipeline in “claim” collection

**Answer:**

db.claim.aggregate([{

$match:{Claim\_status:"Given"}

},{

$group:{

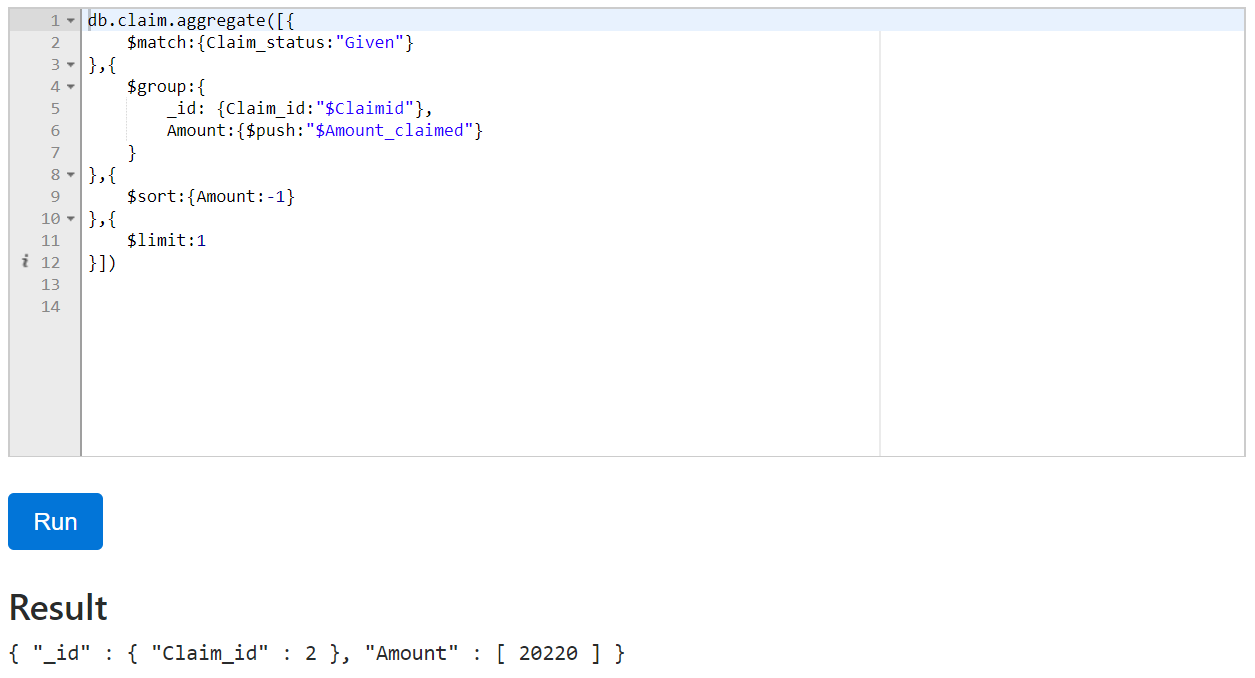
\_id: {Claim\_id:"$Claimid"},

Amount:{$push:"$Amount\_claimed"}

}

},{$sort:{Amount:-1}

},{ $limit:1}])



**MAP-REDUCE PIPELINE**

**Question:** Write a query to find anyone who submitted more than 1 claim using a map-reduce pipeline in “claim” collection

**Answer:**

db.claim.mapReduce(

function(){

emit(this.Insurance\_id, 1);

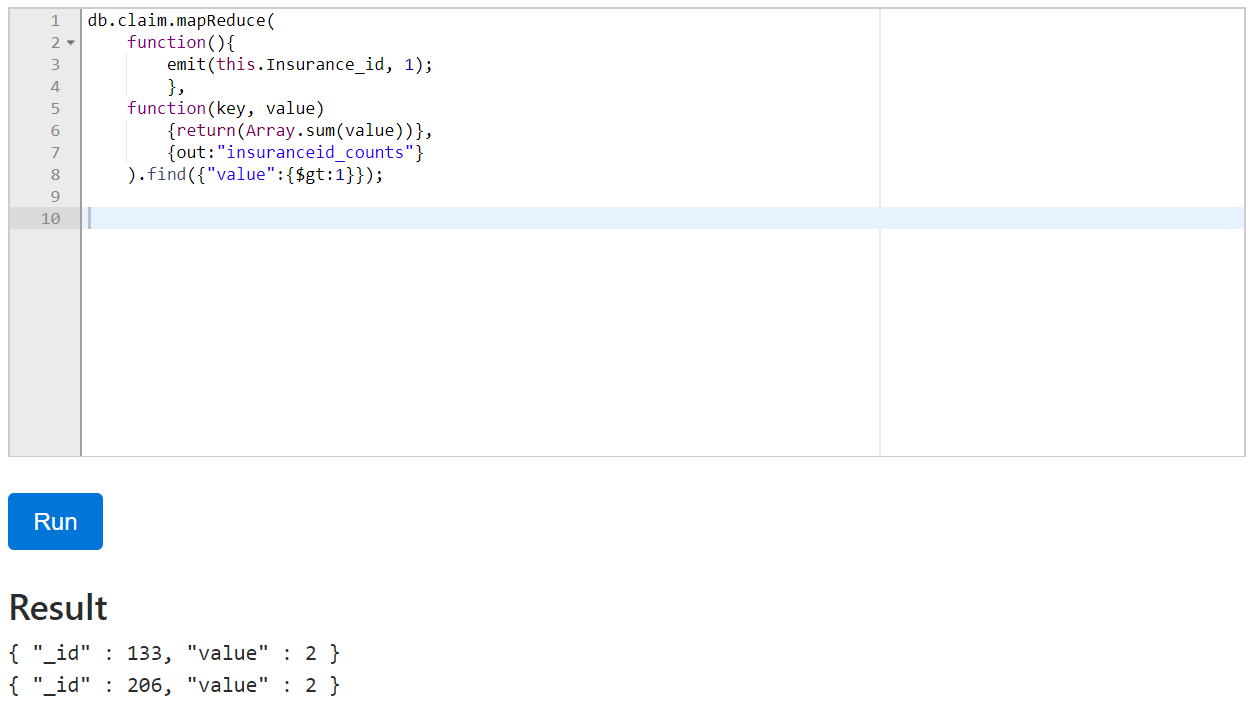
},

function(key, value)

{return(Array.sum(value))},

{out:"insuranceid\_counts"}

).find({"value":{$gt:1}});



**Question:** Write a query to count the number of employee in each department using a map-reduce pipeline in “policyholder” collection

**Answer:**

db.policyholder.mapReduce(

function(){

emit(this.Dept\_id, 1);

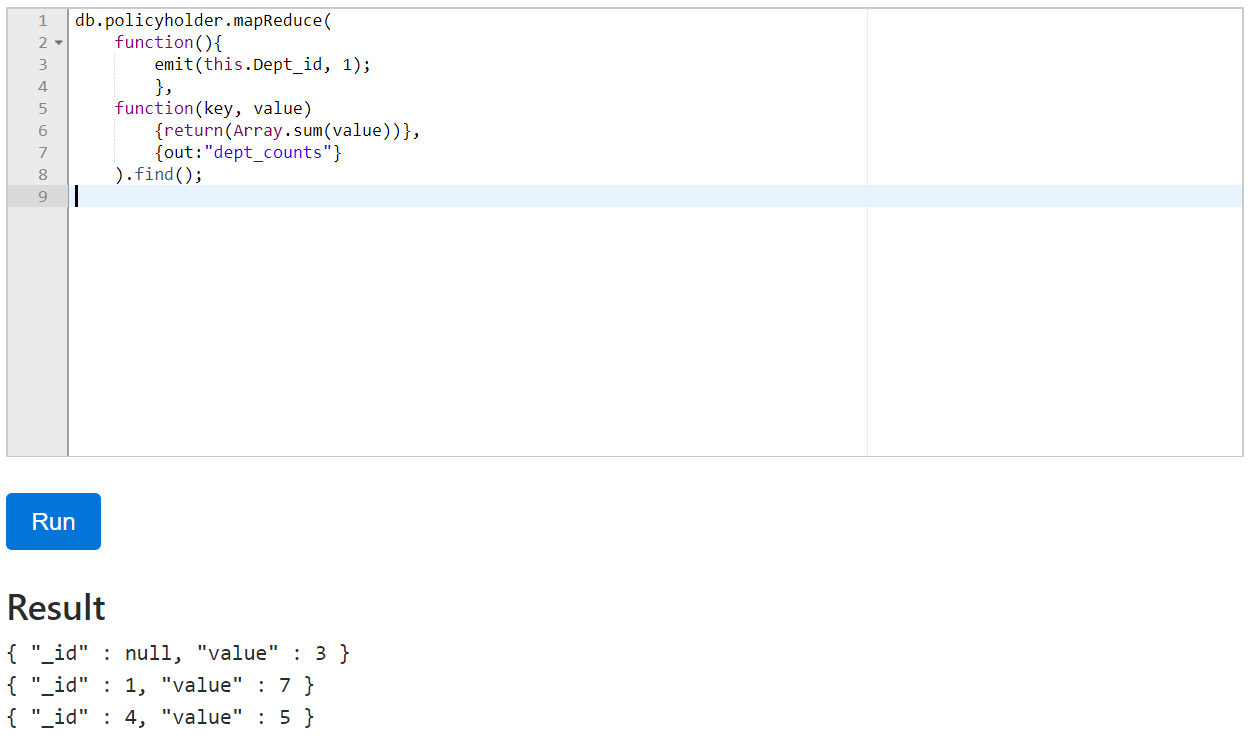
},

function(key, value)

{return(Array.sum(value))},

{out:"dept\_counts"}

).find();



\*null = the policy holder is not an employee in this company