**Pregunta 1 (2 puntos).**

Crear un procedimiento almacenado para que pueda ingresar películas. Se debe validar que no se ingrese películas repetidas. En el caso que se repita, se debe mostrar un mensaje de alerta de película repetida.

CREATE PROC INSERTAR\_MOVIES

@id\_movies int ,

@title varchar (8),

@year int ,

@language varchar (20),

@duration int,

@id\_director int ,

@genre\_id int as

declare @validador int -- validador de resultados

select @validador = count (id) from movies where id = @id\_movies or title= @title

if (@validador<1)

begin

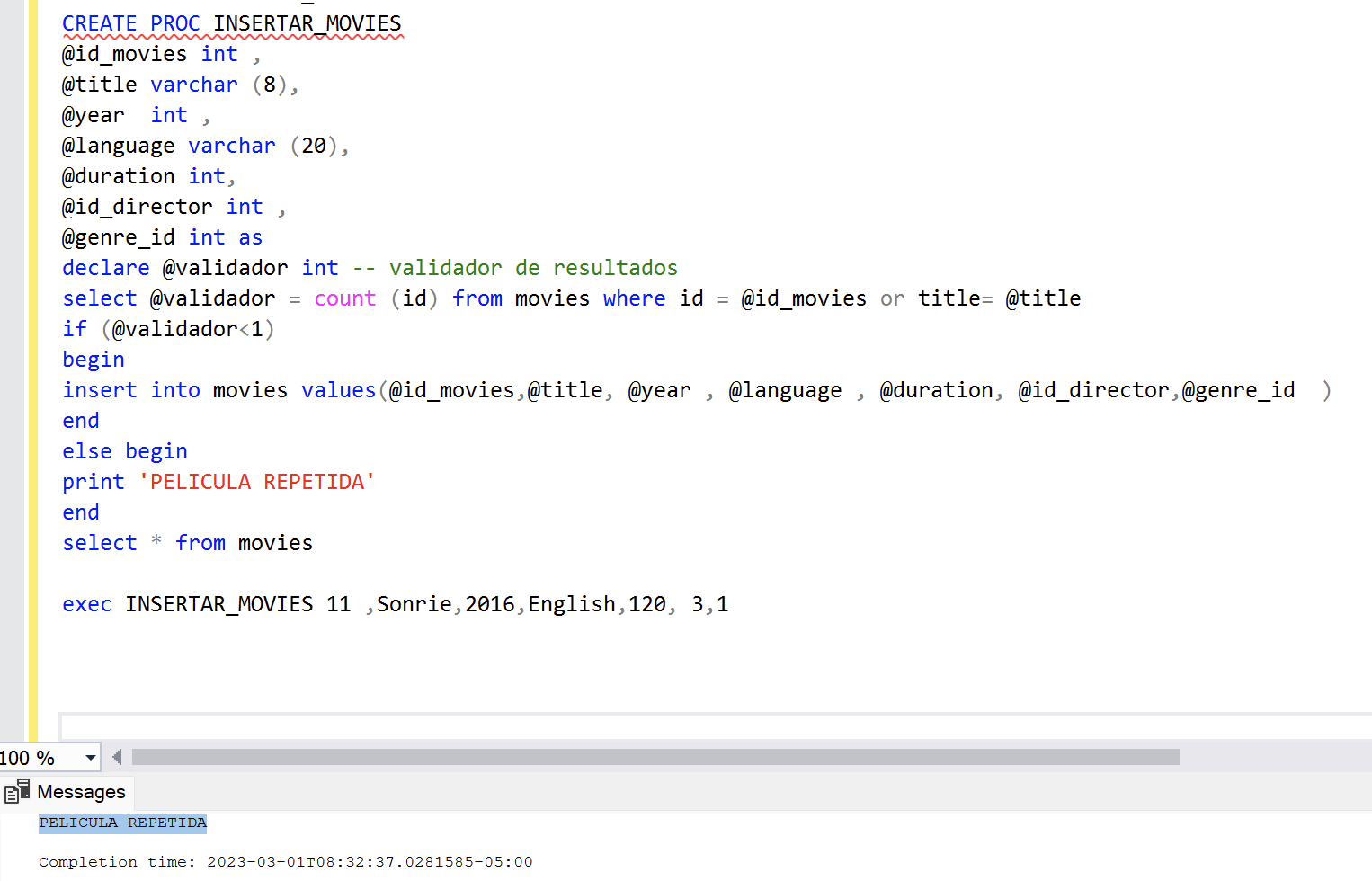
insert into movies values(@id\_movies,@title, @year , @language , @duration, @id\_director,@genre\_id )

end

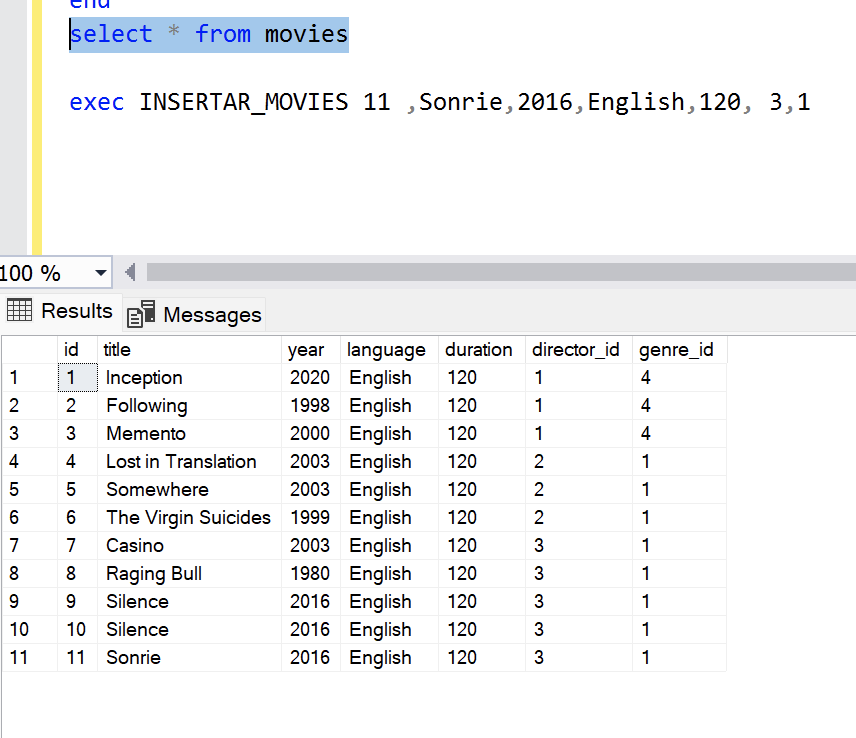
else begin

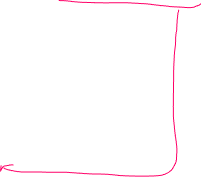
print 'PELICULA REPETIDA'

end

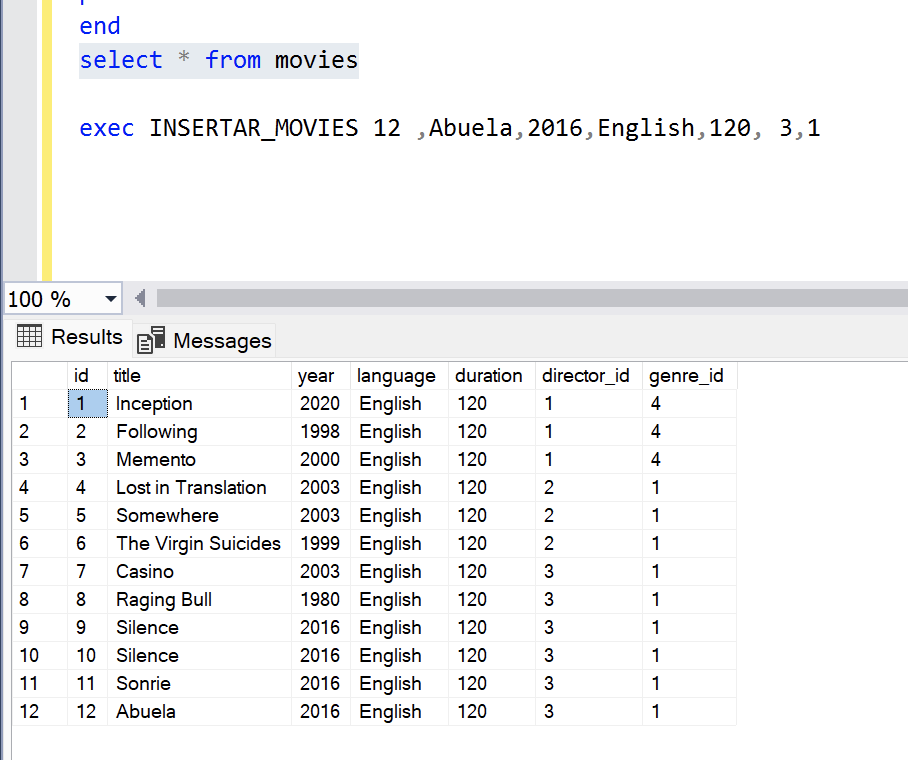


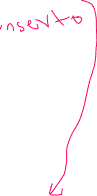






Explication: El procedimiento realiza una validacion de Id y titulo de la tabla de Movies para poder ingresar. Si el id o el titulo ingresado es igual al id existente no podra agregar mas peliculas y nos mostrara un mensaje PELICULA REPETIDA. Solo agrega peliculas cuando son muy diferentes al ID o TITULO de la pelicula.





**Pregunta 2 (4 puntos).**

Crear una vista que retorne los directores que tienen películas con un rating superior a cinco estrellas. Las películas de los directores deben ser de este nuevo siglo. Los campos para mostrar es director ID, nombre del director, película ID, nombre de la película, año y rating de la película.

create function f\_movies\_per\_directores() returns table

as return

select d.id as directorID, d.name as Nombre, m.id as movieID, m.title as Pelicula, m.year as Anio, r.stars as Rating

from directors d

join movies m on d.id=m.director\_id

join ratings r on m.id=r.movie\_id

where m.year >= 2000 and r.stars >= 5

-- MUESTRA LAS PELICULAS CON UN RATING DE 5 ESTRELLAS, PORQUE ES EL MAXIMO RATING,

----SEGUN EL ENUNCIADO DICE MAYOR A 5, PERO

-- 5 ESTRELLAS ES LO MAXIMO QUE HAY, ENTONCES SE IGUALA A 5 ESTRELLAS

create procedure sp\_rating\_per\_directors

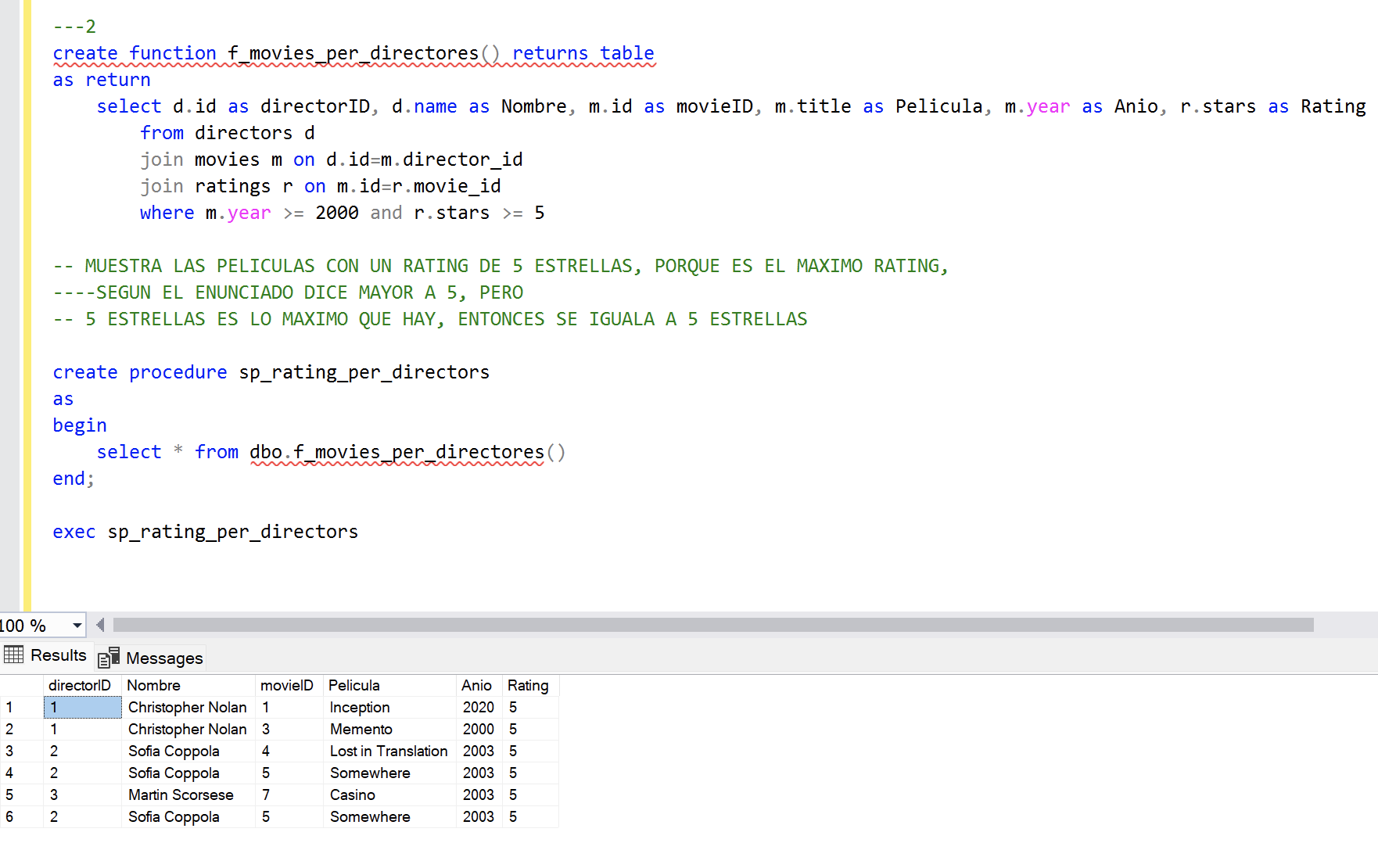
as

begin

select \* from dbo.f\_movies\_per\_directores()

end;

exec sp\_rating\_per\_directors

****

**Pregunta 3 (4 puntos).**

Utilizando la pregunta anterior. Crear un procedimiento almacenado donde todos los actores que han participado en las películas de un determinado director (ingresado como parámetro)

create function f\_actor\_per\_direct(@nombre varchar(40)) returns table

as return

select d.name as Director, a.name as Actor

from directors d

join movies m on m.id=d.id

join movie\_cast mc on m.id = mc.movie\_id

join actors a on mc.actor\_id = a.id

where d.name = @nombre

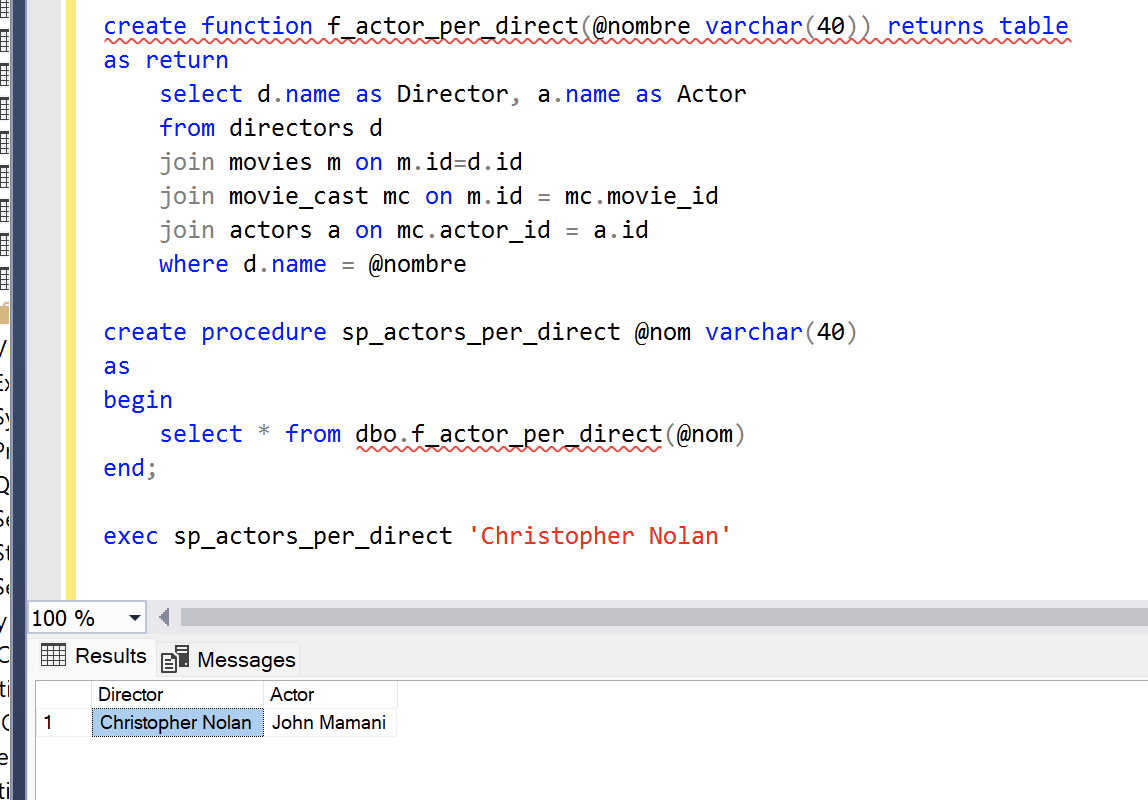
create procedure sp\_actors\_per\_direct @nom varchar(40)

as

begin

select \* from dbo.f\_actor\_per\_direct(@nom)

end;

****exec sp\_actors\_per\_direct 'Christopher Nolan'

**Pregunta 4 (4 puntos) Caso de estudio: Agencia de Viajes**

use('EF\_201919105')

db.createCollection("Pasajes",{

validator:{

$jsonSchema:

{

bsonType: "object",

description: "Information about Pasajes",

required: ["ciudad\_origen","ciudad\_destino","fecha\_hora","nombreAero","NroAsiento","Precio","datosPasajero","hospedaje"],

properties:{

ciudad\_origen:{

bsonType: "string",

description: "La ciudad de Origen debe ser un string y es obligatorio"

},

ciudad\_destino:{

bsonType:"string",

description:"la ciudad destino debe ser un string y es obligatorio"

},

fecha\_hora:{

bsonType:"string",

description:"La fecha debe ser un string y es obligatorio"

},

nombreAero:{

bsonType:"string",

description:"El NombreAero es un string y es obligatorio"

},

NroAsiento:{

bsonType:"int",

minimum:1,

description:"La valoración es un numero decimal mayor a 0 y menor igual que 5 estrellas y es obligatorio"

},

Precio:{

bsonType:[ "double" ],

description:"El precio debe puede ser un numero decimal y es obligatorio"

},

datosPasajero:{

bsonType: "object",

description: "Information about datosPasajero ",

required: ["Apellidos","Nombres","DNI"],

properties:{

Apellidos:{

bsonType:"string",

description: "El apellido es un string y es obligatorio"

},

Nombres:{

bsonType:"string",

description:"Los nombres son string y es obligatorio"

},

DNI:{

bsonType:"string",

description:"El DNI es string y es un dato obligatorio"

}

}

},

hospedaje:{

bsonType: "object",

description: "Information about hospedaje ",

required: ["cantidad\_noches","cantidad\_personas\_habitacion","precio\_por\_noche"],

properties:{

cantidad\_noches:{

bsonType:"int",

description: "El cantidad noches es un tipo dato int y es obligatorio"

},

cantidad\_personas\_habitacion:{

bsonType:"int",

description:"antidad\_personas\_habitacion son int y es obligatorio"

},

precio\_por\_noche:{

bsonType:[ "double" ],

description:"El precio\_por\_noche es double y es un dato obligatorio"

}

}

},

}

}

}

}

)

DATOS: INGRESADOS PARA LA VALIDACION DE COLECCION

use('EF\_201919105')

db.Pasajes.insertOne( {

ciudad\_origen: "Lima",

ciudad\_destino: "Cusco",

fecha\_hora: "23/03/2023 4:00am",

nombreAero: "Avianca",

NroAsiento: 56,

Precio: Double(559.50),

datosPasajero: {

Apellidos: "MAMANI",

Nombres: "John",

DNI:"65656555"},

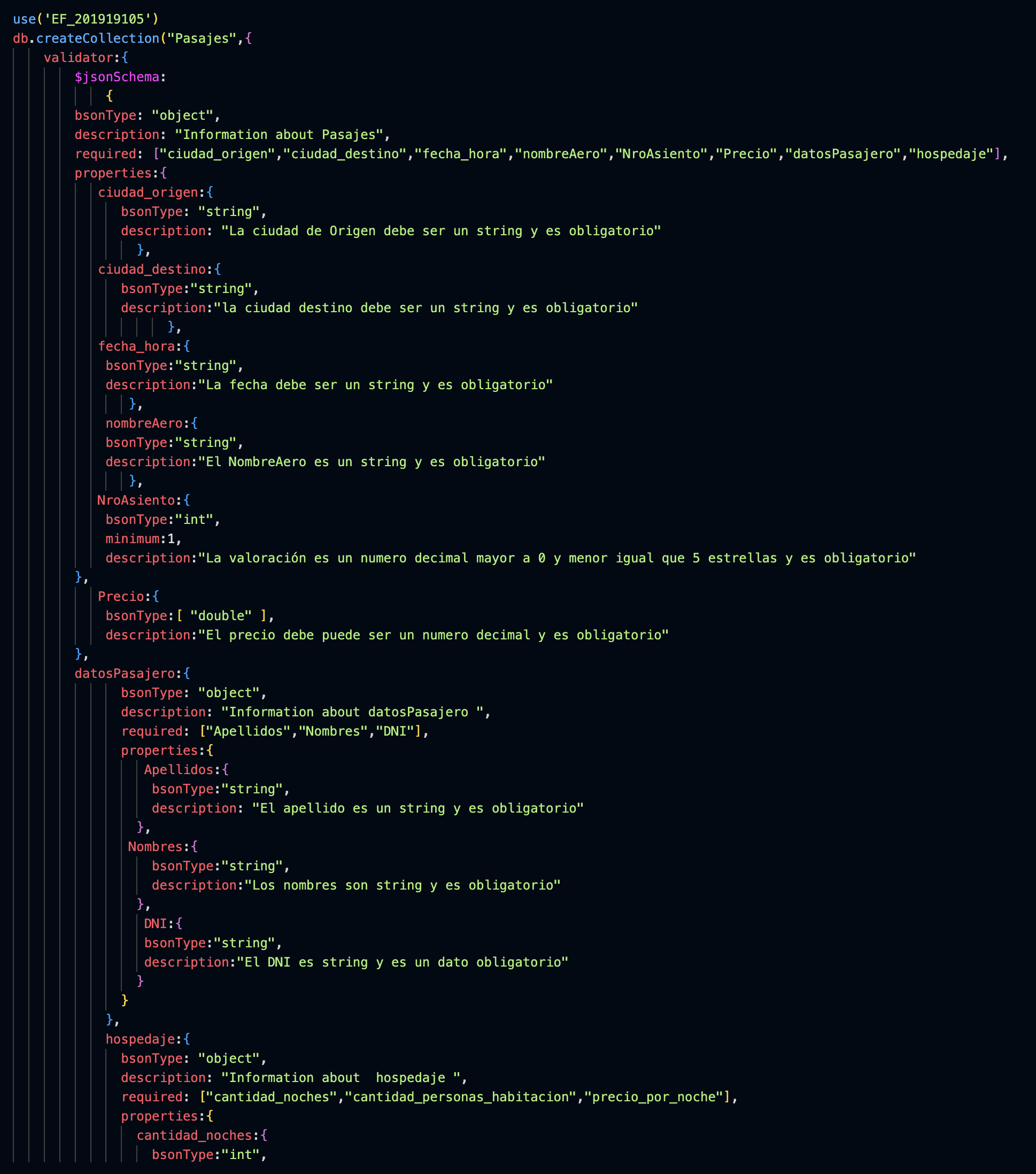
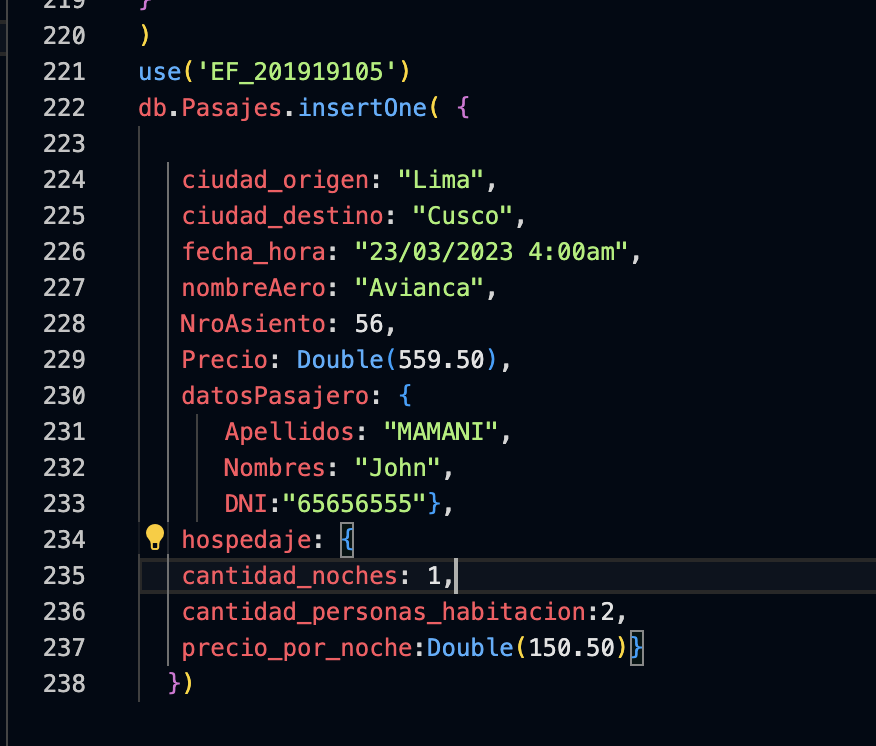
hospedaje: {

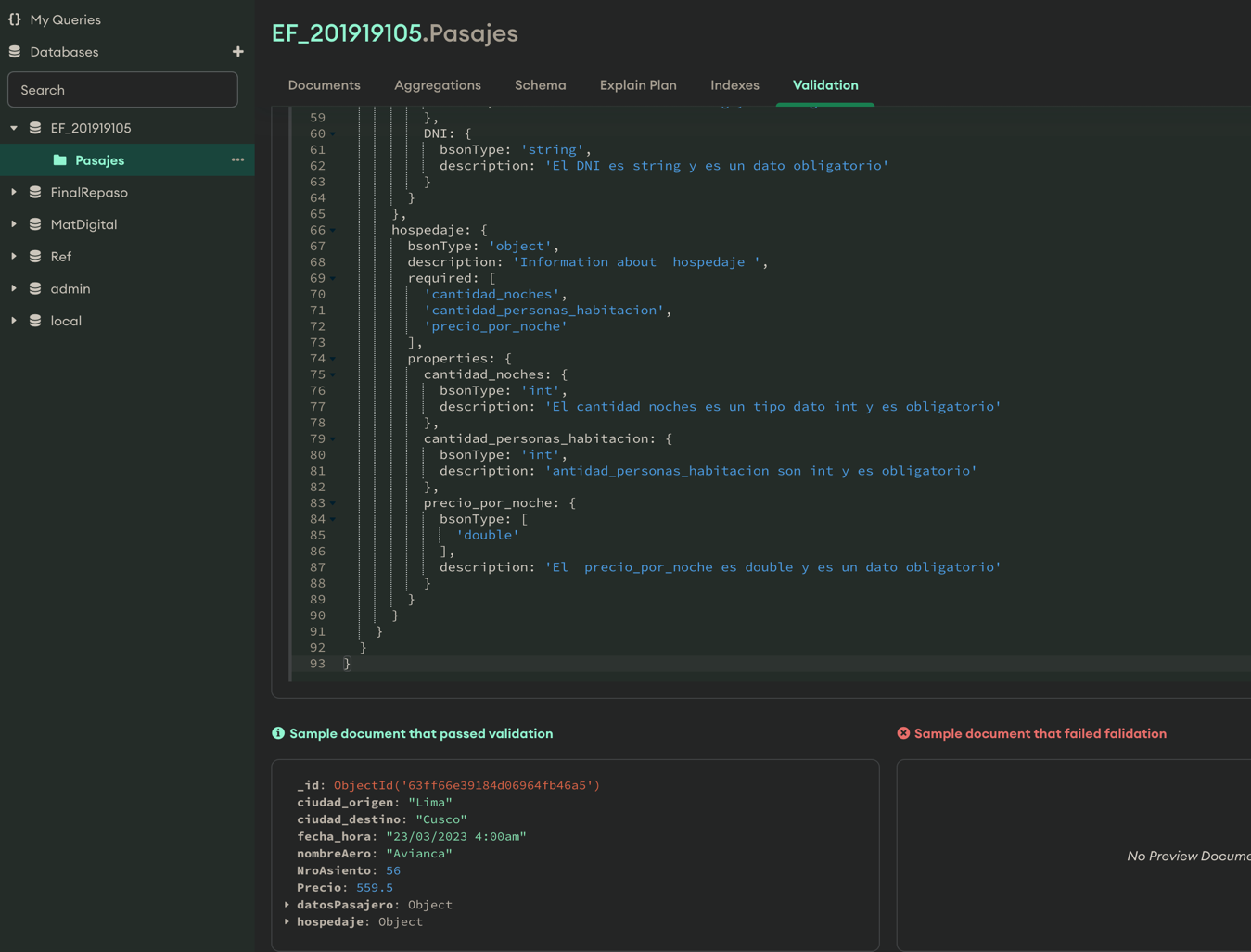
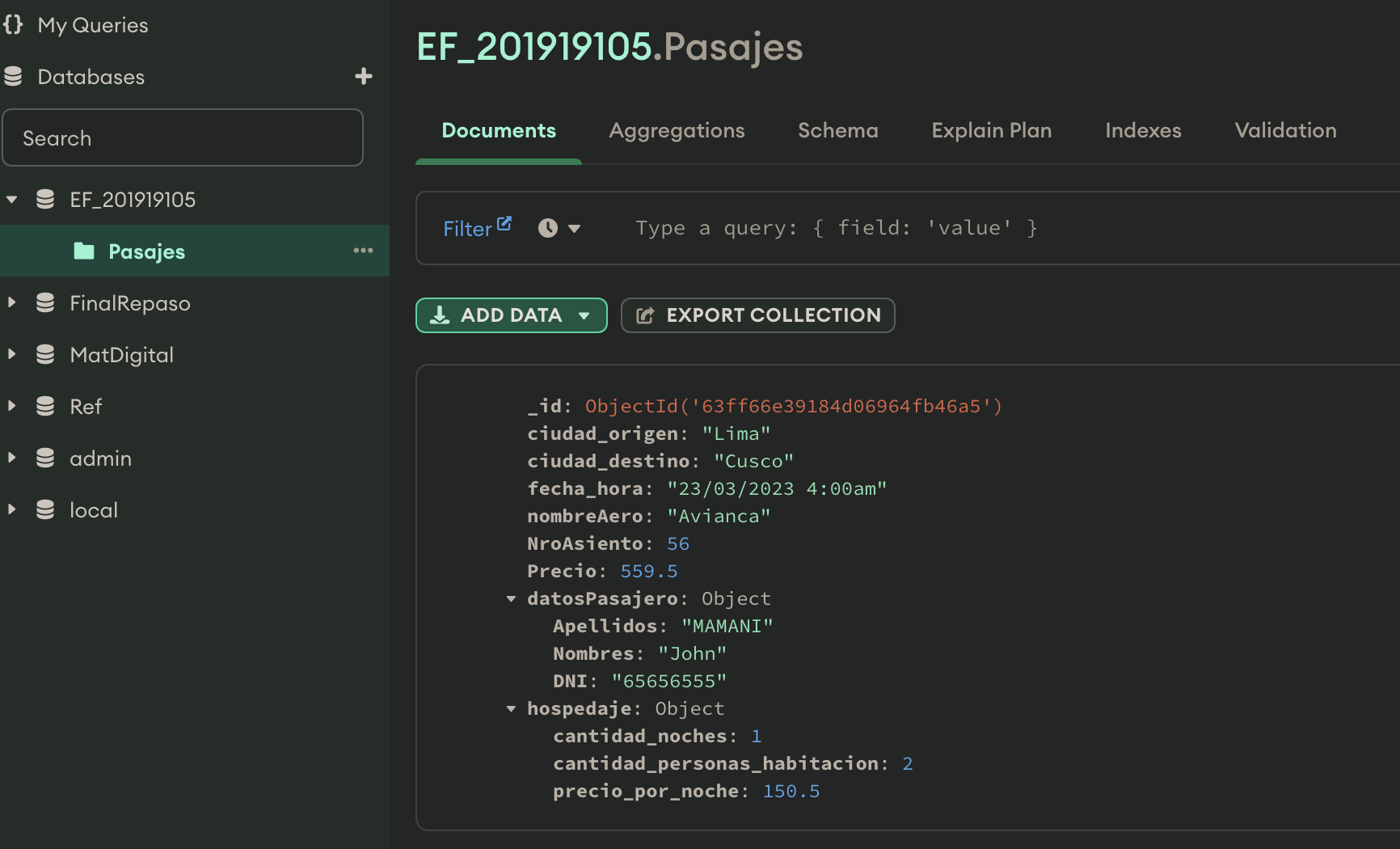
cantidad\_noches: 1,

cantidad\_personas\_habitacion:2,

precio\_por\_noche:Double(150.50)}

})

SE REALIZO LA COLECCIÓN PARA LA VALIDACION EN VISUAL STUDIO CODE Y CONECTADO AL MONGO COMPAS UNA VES VALIDADO SE REVISO SI LOS DATOS FUERON CARGADOS CORRECTAMENTE A MONGO COMPAS

AQUÍ MOSTRAMOS LAS VALIDACIONES EN MONGOCOMPAS

**Pregunta 5 (2 puntos).**

Escribir una consulta que permita mostrar todos los estudiantes de un determinado curso y ciudad.

db.students.Aggregate(

[{$group:{"courses":[], "address":{}}}]

)

**Pregunta 6 (2 puntos).**

Escribir una consulta que permita actualizar a un determinado estudiante cambiándole su año de ingreso.

db.students.updateOne(

{year:2016},

{

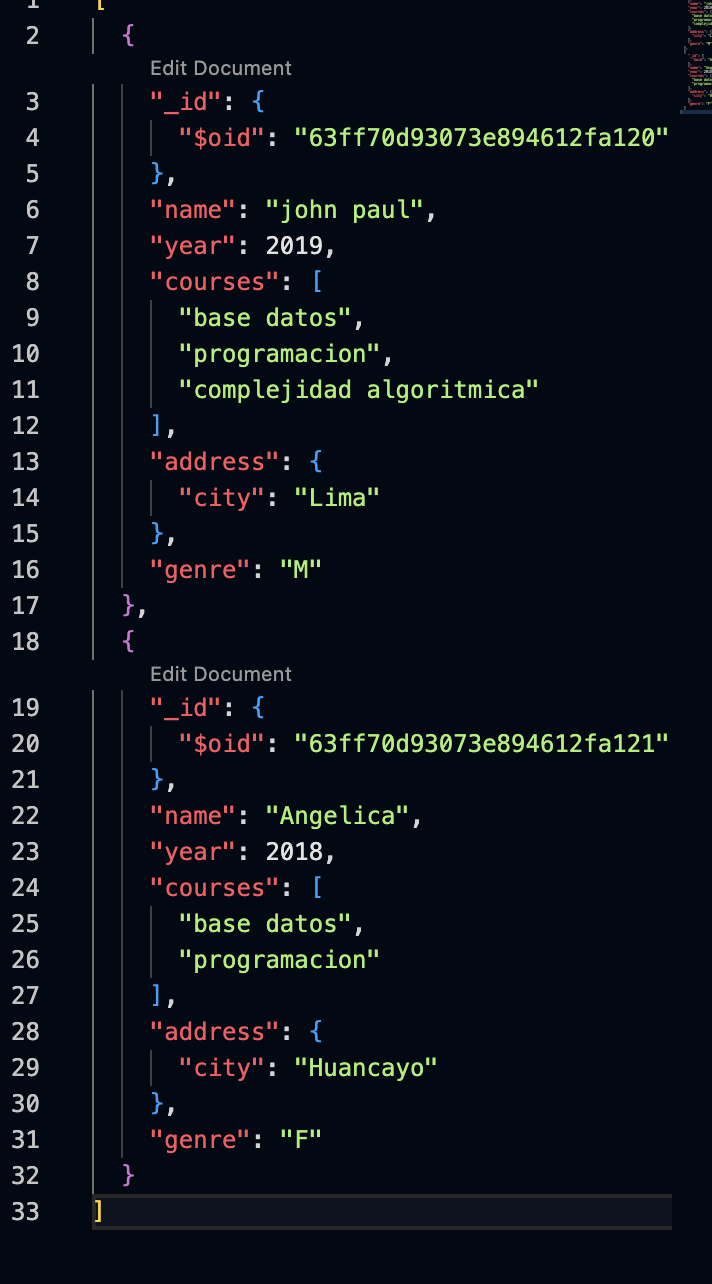
$set:{

year:2017

}

}

)



use('EF\_201919105')

db.student.updateOne(

{year:2019},

{

$set:{

year:2022

}

}

)

**Pregunta 7 (2 puntos).**Escribir una consulta que permita mostrar todos los estudiantes que ingresaron entre el **2015** y **2018** y que sean de la ciudad de **LIMA**.